SEPSIS PREDICTION USING TEMPORAL CONVOLUTIONAL NETWORK

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER AND INFORMATION SCIENCES

Supervisor

Assoc. Prof. Jian Yu

Dr. Sam Madanian

July 2021

By

Kaiyuan Zeng

School of Engineering, Computer and Mathematical Sciences

Abstract

*S*epsis is one of the leading causes of death in hospitals across the world, and it has attracted more and more attention in increasingly aging countries. Every year 5.4 million people worldwide die of sepsis. With the development of social life, predicting sepsis has become more and more important. The main sign of sepsis is multiple organ failure. In 2016, advances in medicine and technology helped redefine the disease standards for this disease. This thesis uses the Sepsis-3 standard to study adult patients. Infected patients with a sequential organ failure assessment (SOFA) score higher than 2 are marked as sepsis patients.

Nowadays, with the advancing at a rapid pace of data mining and artificial intelligence(AI), people's research on the problem of sepsis prediction has become more and more in-depth. This thesis mainly focuses on the prediction of the probability of septicemia among patients in the intensive care unit(ICU). We have developed a deep temporal convolutional network to predict sepsis. At the same time, a machine learning model (decision tree) and a deep learning LSTM model have been developed as the test benchmark model. MIMIC-III is the source database for model development,validation and testing. Our goal is to use 12-hours observational health data to predict whether sepsis will occur in following 6 hours. Our innovation is to mark sepsis with the time of onset instead of the ICD-9 code. The project first used Postgres to extract relevant data from MIMIC-III, and performed data preprocessing, and then established one machine learning model for sepsis prediction, Decision tree and two deep learning

models TCN and LSTM, Decision tree and LSTM model as a benchmark model to verify the performance of the TCN model. The three models are optimized separately. The decision tree uses GridSearchCV to automatically adjust the parameters max_depth, and finally the best max_depth is selected as 5. LSTM and TCN are optimized by setting epochs, the best model is the model with the highest verification accuracy for 20 iterations. Evaluation metrics (Accuracy, Precision, Recall, F1-score, and AUC-ROC) will be used to measure the performance of the model. When predicting sepsis 6 hours before onset on the new reality label, the area under the ROC curve of our proposed TCN model is 0.944, the accuracy is 0.893. The results show that, compared with machine learning methods and LSTM, time convolutional networks converge faster and have better performance. The model is robust and high-precision, and may be used as a tool for hospital sepsis prediction.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of student

Acknowledgements

First of all, I want to thank Auckland University of Technology for giving me a good graduate life, and thank the teachers and classmates for their care and help. I learned a lot in school, which made me go from completely ignorant of programming to being able to complete the project independently. I am very fortunate that I have come to a good environment and have met many mentors and helpful friends. They have provided me with a great deal of direction and assistance so that I may successfully complete my education. I'd like to thank them from the bottom of my heart.

Secondly, I would like to thank my mentors Assoc. Prof. Jian Yu and Dr. Sam Madanian. They are all knowledgeable, rigorous, and humble, and they pay great attention to the academic ability and practical ability of students. They provided assistance on the research path of my thesis, provided me with thorough instruction on the challenges and uncertainties I faced during the writing process, and made several useful recommendations for improvement. They invested a lot of time and effort into my education and thesis writing. At the same time, I am thankful to the professors and all of my classmates for teaching me and assisting me, and we had a wonderful and memorable time together.

Finally, I'd want to express my gratitude to the specialists and academics who examined this work and attended the defense for their time and effort.

Chapter 1

Introduction

This chapter consists of four parts. The first part introduces the background and motivation of this project. The second part puts forward the problems and goals of this project. The contribution of this work will be introduced in the third part. The last part introduces the structure of this thesis.

1.1 Background and Motivation

The original definition of sepsis was proposed in 1991 and was called Sepsis-1 (Bone et al., 1992). Sepsis is defined as the simultaneous presence of systemic inflammation and infection, as a result, the systemic inflammatory response syndrome (SIRS), see the literature review below for details. Sepsis-1 was updated to Sepsis-2 through the expanded diagnostic criteria list in 2001 (Levy et al., 2003), but no alternative was provided, so the definition of sepsis has not altered. In 2016, a new definition of sepsis, Sepsis-3, was proposed, which describes sepsis as a life-threatening organ malfunction induced by an imbalance in the host's reaction to infection (Rather & Kasana, 2015). In Sepsis-3, patients with infection quantified by SOFA increase ≥ 2 are defined as sepsis (Vincent et al., 1996).

Every year, an estimated 31.5 million individuals worldwide contract sepsis, and sepsis has become one of the major causes of mortality among Intensive Care Unit (ICU) patients (Vincent, Marshall, Ñamendys Silva & François, 2014). In the United States,10% of patients admitted to the ICU have sepsis, and roughly 25% of ICU beds are filled by sepsis patients (Pires, Neves & Pazin-Filho, 2019). The benign stage of sepsis has a death rate of 17%, whereas the severe form has a mortality rate of 26% (Fleischmann et al., 2016). Sepsis is an expensive disease in the ICU. In 2013, it cost 23,663$ millions in the United States, accounting for 6.2% of the cost of all hospitals in the United States (Celeste, Torio & Brian J. Moore, 2013), and increased to 30320$ millions, accounting for 8.8% of the cost in 2017 , and it continues to grow (Celeste, Torio & Brian J. Moore, 2017).

Given the high death rate of sepsis patients in the ICU, the earlier the risk of sepsis among ICU patients may be identified, the better. Early and precise detection of patients at high risk of sepsis can assist ICU clinicians in making the optimal clinical decisions, resulting in better clinical outcomes. Early therapy has been proven to have a substantial favorable influence on survival in studies (Angus et al., 2001; Kumar et al., 2006), in particular, relevant verification shows that for every hour of delay in treating patients with sepsis, the mortality rate will increase by 7.6% (Nguyen, Corbett, Steele & Banta, 2007). Most of the research now focuses on the prediction of septic shock. Septic shock is already an aggravation of sepsis and has seriously threatened the lives of patients. It is more important to detect sepsis early and predict whether the patient is likely to develop sepsis.

So far, many models for predicting sepsis have been developed. The earliest early warning of sepsis is based on machine learning, (Thiel et al., 2010; Calvert et al., 2016; Desautels, Calvert, Hoffman & Jay, 2016). Due to the recent development of artificial intelligence, many authors have turned to research on the application of neural network methods in the direction of sepsis prediction (Kam & Kim, 2017; Raghu, Komorowski

& Singh, 2018). Compared with the machine algorithm, the performance of neural network algorithm in predicting sepsis has a very obvious improvement. For example, the logistic regression model based on MIMIC-II (Calvert et al., 2016), such as the gradient tree boosting model based on non-MIMIC databases (Mao et al., 2018) and the supervised machine learning model of gradient boosting (Delahanty, Alvarez, Flynn, Sherwin & Jones, 2019). Another example is the non-MIMIC-based recurrent neural network (Futoma et al., 2017). However, most of them still use the MIMIC-III data set (Raghu et al., 2018; Desautels et al., 2016). For a long time, Recurrent neural networks (RNN) and Long Short Term Memory networks(LSTM) have long been the major approaches for dealing with prediction issues in deep learning. LSTM can significantly solve the problem of gradient disappearance as compared to RNN.

Temporal Convolutional Network (TCN) is a new type of neural network with a fully convolutional structure, which has been proven to be superior to recurrent networks. And deep learning has been proven to be superior to traditional machine learning algorithms. TCN puts forward two principles that must be observed. The first is whether the convolutional structure suffers from spesis at the calculation time t, which can only involve the feature input before the time t, and there will be no "information leakage". The other is that the length of sequence data is relatively loose. For spesis data, patients have more feature points. We can choose the appropriate feature amount according to our needs. When we need to expand in the future, the model does not need to be modified. It turns out that TCN is very suitable for processing serialized data because they only use previous data to generate new data. In addition, SOFA, as an approximation of the starting time, can be used for the development of time window forecasts. Therefore, TCN is very suitable for predicting whether sepsis will occur in the next 6 hours.

1.2 Research Questions and Goals

The goal of this thesis is to use 12-hours observational health data to predict whether sepsis will occur in following 6 hours. Use ICU data on MIMIC-III.

This thesis establishes Decison Tree model(ML), LSTM model(deep learning) and TCN model(deep learning). The benchmark model includes one machine learning model and one deep learning LSTM model. The decision tree is chosen here because it is an effective and mature algorithm often used by managers and analysts. It is a commonly used model in machine learning and can intuitively display the entire decision-making process. The decision tree may not represent the optimal performance of the entire machine learning algorithm, but it is simple to implement, is also one of the commonly used machine algorithms, and can well promote the progress of our experiment. Another deep learning model, LSTM, is chosen here because LSTM is an excellent variant of RNN, inheriting most of the characteristics of RNN models, and at the same time, it solves the vanishing gradient problem caused by the gradual reduction of the gradient backpropagation process. It has long Time memory function is very suitable for dealing with timing problems, so to show the advantages of TCN, we choose LSTM, which has been proven to be a very good model as a benchmark.

The main focus of this thesis is the prediction effect of the TCN model. And improvements have been made to three models to improve the prediction performance.

The research question is divided into four parts:

1. What features from dataset should be considered?
2. How do we use the three AI methods of TCN, ML, and LSTM to predict sepsis?
3. How to validate the predictive models and select the best performing model in each model.
4. Which algorithms are effective and can meet the evaluation critetia?

The full process of conducting this research is seen in Figure 1.1.

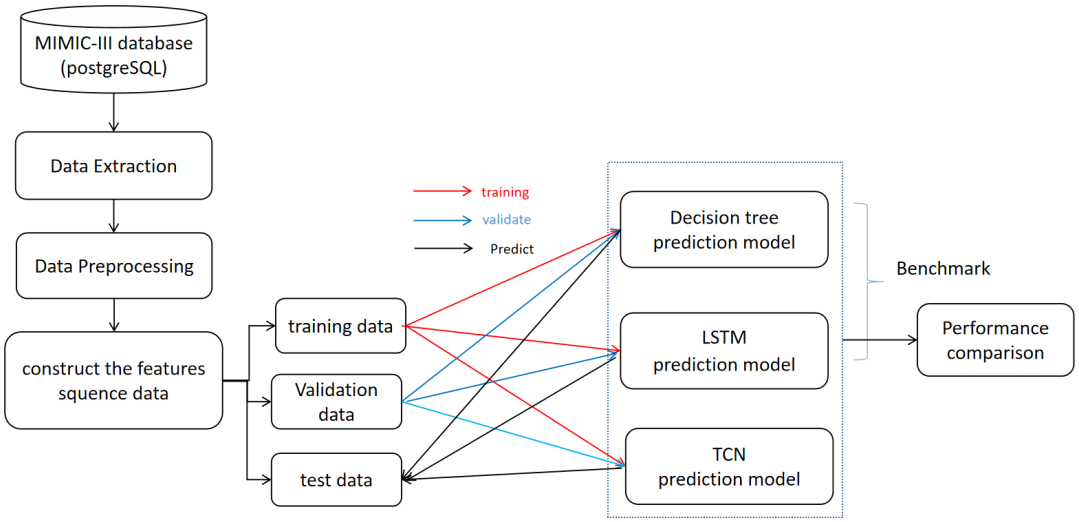

Figure 1.1: The full Process of conducting this research

1.3 Research Contributions

This thesis is based on `TCN` training prediction, using Python language, and the implementation content is as follows:

- Mark sepsis with the time of onset instead of the ICD-9 code.
- Data preprocessing in scala and python pandas through apache spark.
- Use 12-hours observational health data to predict whether sepsis will occur in following 6 hours, using the time dependence of sepsis.
- Created one machine learning models and two deep learning models, and use cross-validation method or epochs method to obtain better performing models.
- Use pytorch to create a TCN-based predictive sepsis model. After scikit-learn evaluation, it has excellent predictive performance compared with related machine learning methods and LSTM.

1.4 Research Benefits

This research will benefit ICU clinicians and patients who may suffer from sepsis.

- For ICU clinicians: Early and accurate detection of patients at high risk of sepsis can help ICU clinicians make the best clinical decisions and obtain better clinical results.
- For patients who may suffer from sepsis: early and accurate detection of sepsis in the next few hours can obtain early prevention, early treatment, prevent the deterioration of the condition, prevent the development of septic shock, and improve the survival rate of patients.

1.5 Thesis Structure

Deep learning technology automates the process of feature extraction and selection, making categorization easier. Therefore, deep learning models are increasingly used to detect various diseases (Acharya et al., 2018; Yildirim, Baloglu, Tan, Ciaccio & Acharya, 2001; Oh et al., 2020). In this study, we used a temporal convolutional network to predict sepsis. The method we proposed can not only predict sepsis quickly, but also has high accuracy. This thesis has developed two deep learning models, LSTM and TCN. In order to verify the effectiveness of the deep learning model, a machine learning model decision was also developed Tree classifier.

Around the above, this thesis is divided into six chapters, each of which is as follows:

Chapter 1 Mainly introduces the research background of the subject, the main research issues, research goals and results. Finally introduce the thesis organizational structure.

Chapter 2 It is mainly a literature review, reviewing the related literature on sepsis prediction in the past, and related applications based on deep learning prediction, as well as the introduction and application of MIMIC-III. Then introduced the relevant theoretical knowledge to built decision tree, LSTM and TCN models, and the theoretical knowledge of model evaluation indicators.

Chapter 3 This chapter first introduces the entire thesis research process. Then mainly generates three data sets for model training, verification and testing. First introduced two codebases used, one is Spesis-mimic, the other is MIMIC-III. Then it introduces how to extract sepsis-related data sets from PostgreSQL, and process the data sets, and finally split them into training set, validation set and test set according to the ratio of 7:1:2. Finally, the characteristic sequence data set of the industry is constructed.

Chapter 4 It mainly builds three models of decision tree, LSTM and TCN. Main implement related models to predict sepsis. First implemented decision tree based on python's sklearn.tree.DecisionTreeClassifier, and adjusted the parameters through GridSearchCV. Then for the Deep learning models LSTM and TCN, implement related models, and use epochs to select the best model.

Chapter 5 First of all, the performance of the three models is compared, and the focus is on the performance comparison between TCN and the benchmark model. Then Mainly discuss the content of this thesis research, and analyze the benefits and limits.

Chapter 6 Summarize and look forward to the future. This chapter summarizes the work of this thesis, and proposes future research directions in this field.

Chapter 2

Literature Review

In this chapter, we review the research background in sepsis and its prediction by consulting the literature, studied the development of deep learning in predicting time series data, and studied the application of RNN, LSTM, TCN and other deep learning methods in predicting sepsis. Finally, we introduced MIMIC-III database and its historical application. And introduced the decision tree classifier we will implement, the relevant theoretical knowledge of LSTM and TCN models, and the relevant indicators of model evaluation.

2.1 What is Sepsis?

Sepsis is a clinical syndrome caused by pathogenic bacteria and their toxins invading the bloodstream. Pathogens are usually bacteria, but can also be fungi or mycoplasma. Mortality is the main cause of death in ICU (Bone et al., 1992). Despite its high mortality rate, the cause of sepsis is still unclear. In recent years, people have gained a new understanding of the pathogenesis of sepsis and the criteria for judgment. In the study of sepsis, more and more attention is paid to the body's systemic response to invading microorganisms and their toxins (Vincent, 1997).

In 1991, the American College of Thoracic Physicians and the Crisis Care Society defined sepsis as a systemic inflammatory response caused by various infections. The diagnosis conditions are: there is clinical evidence of infection and evidence of systemic reaction caused by infection (Gül, Arslantaş, Cinel & Kumar, 2017). Systemic reaction includes at least the following 2 items:

Table 2.1: Sepsis-1 symptoms

SIRS (Spesis-1)
† **1:** Body Temperature > 38 °C or < 36 °C
† **2:** Heart rate > 90 beats per minute
† **3:** Respiratory rate > 20 breathes per minute or carbon dioxide partial pressure (PaCO2) < 32 mmHg(4.3 kPa)
† **4:** White blood cell count(WBC) > 12,000 cells per mm3 or < 4,000 cells per mm3

According to spesis-1 in 1991, when a patient shows two or more of the above clinical manifestations, the systemic inflammatory response syndrome (SIRS) can be diagnosed, and the patient can be judged to have spesis.

But, SIRS is too broad and has low accuracy and high false alarm rate due to lack of specificity, and some of the characteristic manifestations of sepsis syndrome are not based on the onset of critically ill patients and clinical epidemiological studies. The four indicators of SIRS can only reflect the general severity of the disease, and cannot be used as specific markers for the expanding inflammatory response in the body. Sepsis-1 was updated to Sepsis-2 in 2001 (Levy et al., 2003). It points out that inflammatory response parameters, hemodynamic parameters, organ dysfunction, and tissue perfusion parameters are four infection parameters, and if two or more of them are satisfied, it is diagnosed as sepsis. However, due to the complexity of diagnostic indicators and difficulty in clinical application, the spesis-1 standard is still being used.

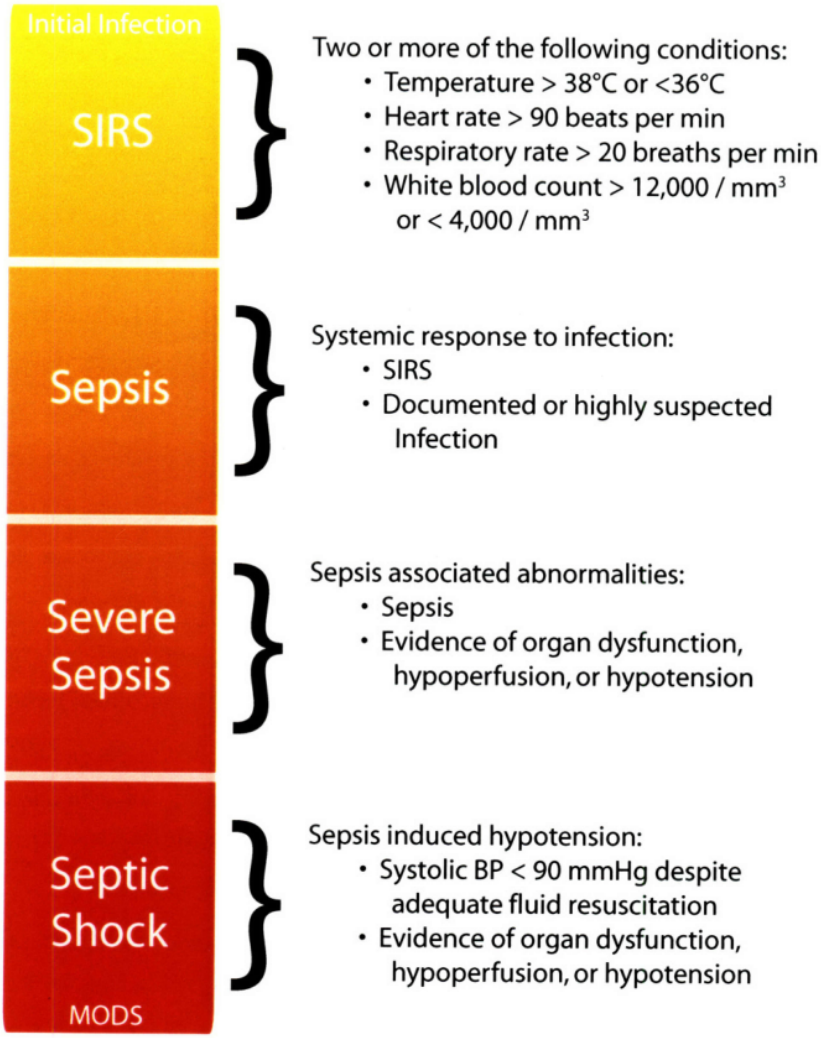

Figure 2.1: The clinical spectrum of sepsis

In 2016, a new definition of sepsis, Sepsis-3, was proposed, which redefines the disease as a life-threatening organ dysfunction caused by the imbalance of the host's response to infection (Rather & Kasana, 2015).

When sepsis is further complicated by organ dysfunction or abnormal hypoperfusion, severe sepsis will occur (Bone et al., 1992). A subset of severe sepsis is septic shock (Otero et al., 2006). The clinical spectrum of sepsis is shown in Figure 2.1. It starts from the initial toxic injury, gradually progresses to the stage of increased inflammation,

and develops into a process of septic shock.

2.2 Sepsis Prediction

Sepsis prediction is very important, timely prediction of sepsis can save patients' life. Looking at the literature, it can be found that most of the research on sepsis is focused on the hypothesis that sepsis has developed, and then studying the probability of patient death or the pattern of further deterioration, that is, to deal with the problem of septic shock. Septic shock is a subgroup of sepsis research in which the underlying circulatory and cellular abnormalities are severe enough to increase mortality considerably (Singer et al., 2016). Related work mainly includes exploring survival models (Henry, Hager, Pronovost & Saria, 2015) and hidden Markov models (Ghosh, Li, Cao & Ramamohanarao, 2017). The Septic Shock Early Warning Model (EWS) was created by applying multivariable logistic regression to the features such systolic blood pressure, pulse pressure, heart rate and temperature etc. This model can predict the onset of septic shock with high accuracy one hour in advance (Shavdia, 2007).

For the early prediction of septic shock, some previous studies used a multivariate logistic regression model. A model for predicting septic shock developed by the regression analysis method used by Hug. The data set is 7048 hospitalized patients with hundreds of candidate variables (including vital signs, long-term trends, baseline deviations, etc.). The variables of the final model were selected through cross-validation and other methods, and verified on the data of 3018 patients , The Receiver Operating Characteristic/Area Under the Curve(ROCAUC) value of the model is 0.880, this means that the probability that the classifier can accurately identify patients with sepsis is greater than 88% of patients who do not suffer from sepsis, so the prediction performance of the model is very good (Hug, 2009). Another report also established a multiple regression model, which data is selected in MIMIC-II, which variables

retain hemodynamic data, laboratory test results data and some clinical information. It studied the mortality of 23 septic shock patients over the next 7 days (Carrara, Baselli & Ferrario, 2015). Toma et al. used a logistic regression model to predict the mortality of patients with sepsis (Toma, Abu-Hanna & Bosman, 2007). The model relies on data collected 24 hours before the patient enters the kindergarten and Sequential Organ Failure Assessment (SOFA) score data. The model shows that the performance of the logistic regression model that only uses admission information is better (Toma et al., 2007). In addition, some previous studies used regression tree models to predict the risk of septic septic shock in more than 13,000 non-ICU patients (Thiel et al., 2010). Vieira et al. proposed an improved particle swarm optimization method for feature selection and optimization of SVM parameters to improve the mortality prediction performance of patients with sepsis (Vieira, Mendonça, Farinha & Sousa, 2013). They chose a growing trapezoidal basis function network architecture for the metric variables generated by classification and rules. They mainly have a detailed understanding of the various indicators of patients to improve the predictive performance of patients with sepsis and septic shock (Paetz, 2003). The Septic Shock Early Warning Model system uses a data set of 250 patients with sepsis, of which 65 patients with sepsis have developed septic shock. The data set of this system extracts patient characteristics from the MIMIC database. The model can predict the development of septic shock in patients with sepsis one hour in advance. The AUCROC of the model is 0.928, and the predictive performance of this model is still very high, this means that the probability that the classifier can accurately identify patients with sepsis is greater than 92.8% of patients who do not suffer from sepsis. Another study used recursive partition and regression tree (RPART) analysis to develop and predict a model for non-ICU patients who developed septic shock a few hours before admission to the ICU (Thiel et al., 2010). The data it uses are those of patients who are hospitalized but not in the ICU. The model correctly identified 0.55 septic shock patients. However, the practical significance of

studying septic shock is not great, because the time of septic shock is relatively rare in clinical practice.

We know that septic shock is already an aggravation of sepsis, which has seriously threatened the lives of patients. Therefore, it is more important for us to detect sepsis early, or predict that the patient is likely to develop sepsis, and take measures to prevent the patient from deteriorating to septic shock. In the study of neonatal sepsis, continuous heart rate characteristic monitoring was developed to help the early diagnosis of sepsis in premature infants in neonatal ICU, which has been proved to be effective in the early diagnosis of sepsis (Moorman, Lake & Griffin, 2006). Griffin studied heart rate variability before neonatal sepsis, and finally proved that detecting abnormal heart rate characteristics is effective in the early diagnosis of sepsis (Griffin & Moorman, 2001). However, it is very unfortunate that this detection method did not extend to adults. Although some markers provide reasonable efficacy in predicting sepsis in adults, none or a few can provide early warning of sepsis.

Many literature methods for predicting sepsis are based on machine learning methods. Calvert et al. proposed a sepsis warning algorithm, InSight, which can predict the occurrence of sepsis at least 3 hours before the onset of the first 5 hours of SIRS (Calvert et al., 2016). Another paper predicts sepsis also using a machine learning classification approach called InSight. It uses multivariate data, including vital signs, peripheral capillary oxygen saturation, Glasgow coma score and age, but it is based on Spesis-3 defined to screen sepsis, the model still has a good predictive effect on sepsis even when data is randomly lost (Desautels et al., 2016). Another document uses supervised machine learning called gradient boosting to predict sepsis. The difference is that it uses Rhee clinical monitoring criteria to screen for sepsis (Delahanty et al., 2019). Another study used recursive partition and regression tree (RPART) analysis to develop and predict a model for non-ICU patients who developed septic shock a few hours before admission to the ICU (Thiel et al., 2010).

Recently, with the development of artificial intelligence neural network algorithms, which have powerful algorithms and better performance, many authors have turned to study the application of neural network methods in the direction of sepsis prediction. One study was based on reinforcement learning to explore better treatments for sepsis (Raghu et al., 2018). Another study is to develop early detection of sepsis based on a deep learning model and compare its performance with the Insight model (Kam & Kim, 2017). Gwadry et al. used the decision tree method to analyze the risk of patients suffering from sepsis. He optimized the model by using a multimodal analysis method. The final model achieved a prediction accuracy of nearly 100% for 20 clinical variables (Gwadry-Sridhar, Lewden, Mequanint & Bauer, 2009). Ho et al. believe that the risk of sepsis is hidden in clinical measurement indicators, such as gene expression levels. It uses a combination of missing value interpolation technology and fuzzy prediction model to improve the accuracy of early warning of sepsis (Ho, Lee & Ghosh, n.d.). Tang et al. developed the SVM model to classify patients with sepsis into severe sepsis and SIRS patients. He used principal component analysis (PCA) to reduce dimensionality. The feature space of SVM uses the first three principal components. Finally, the model obtained a prediction accuracy rate of 84.62% (Tang et al., 2010). Lukaszewski et al. proved the effectiveness of using blood sample measurement and miRNA expression levels as training features. He designed a learning multilayer perceptron model to predict the risk of patients suffering from sepsis. The final model achieved 83% prediction accuracy (Lukaszewski et al., 2008).

Although many scholars have conducted research on predicting sepsis, and some scholars have used deep learning algorithms to predict sepsis, there are few literatures on the use of TCN to predict sepsis. This article wants to explore whether the TCN algorithm has advantages over machine algorithms and LSTM algorithms using the same data set.

2.3 Artificial Intelligence Method to Predict Sepsis

2.3.1 Research on TCN to Predict Sepsis

It is worth noting that many literature models use averaging or forward and backward interpolation for the processing of missing values (Calvert et al., 2016). This conversion will produce data artifacts. The absence of data in the medical sector is a deliberate decision made by experts, implying underlying assumptions about the patient's condition.

At present, the two famous TCN models for sepsis prediction are MGP-TCN (*BorgwardtLab/mgp-tcn*, 2021) and MGP-AttTCN (Margherita, 2021). MGP is Multitask Gaussian Process MTGP. AttTCN is attention TCN. The probability distribution is used by the Gaussian process to represent previous knowledge about the function output, and builds a model in the functional space. Based on the correlation between the data, the covariance function is constructed and calculated by Bayesian inference. MTGP is used to handle the situation where the GPR model has multiple outputs. MTGP was first proposed in the literature (Williams, Bonilla & Chai, 2008), and another literature proved the superiority of MTGP in multivariate time series analysis (Dürichen, Pimentel, Clifton, Schweikard & Clifton, 2015). Attention is proposed to solve machine translation, which is the translation of text by a computer, with no human involvement (Bahdanau, Cho & Bengio, 2016).

At present, attention model has become a very important research field in neural network research. The core idea of attention is to weight all the outputs of the encoder and input them into the decoder at the current position to affect the output of the decoder. By weighting the output of the encoder, it is possible to use more context information of the original data while realizing the alignment of the input and output (Woo, Park, Lee & Kweon, 2018).

As shown in Figure 2.2(a), A conventional sequence-to-sequence paradigm has two parts: encoding and decoding. The encoder usually uses a RNN structure to encode the input $\{x_1, x_2, ..., x_T\}$ of the sequence to generate a collection of fixed-length encoding vectors $\{h_1, h_2, ..., h_T\}$. The decoder also uses an RNN structure to read a single input h_T and then output one by one to obtain an output sequence $\{x_1, x_2, ..., x_{T'}\}$. Where T represent the length of the input sequence, and and T' represent the length of the output sequence. The hidden states of the encoder and decoder at each location are represented by t, ht, and st, respectively. This encoding-decoding structure has two main drawbacks. The encoder must compress all input data into a fixed-length vector h_T for the first reason. When this basic fixed-length encoding is used to represent longer and more complicated inputs, the input information is frequently lost.Secondly, such a structure cannot model the correspondence between the input sequence and the output sequence, and this relationship is crucial in tasks like machine translation and text summarization. In a sequence task, each position of the output sequence may be impacted by a particular position of the input sequence, intuitively speaking. However, the classic decoding structure does not consider this correspondence when generating output.

Figure 2.2(b) depicts the framework of the model that introduces the attention mechanism. The function of the attention module is to automatically learn the weights α_{ij}, the goal is to capture the correlation between the hidden state of the encoder h_i and the decoder s_j. The learnt attention weights will be utilized to create a context vector c, which will be sent into the decoder as an input.The context vector c_j is generated at each location of the decoder j by weighted summing of the hidden states of all encoders by the attention weight, namely: $c_j = \sum_{i=1}^{T} \alpha_{ij} h_i$. Therefore, this context vector really offers a way for the decoder to access the whole input sequence and pay attention to certain important places in the sequence. We name this system the attentiveness mechanism.

A study on sepsis is to input the latent function of the Gaussian process into a deep

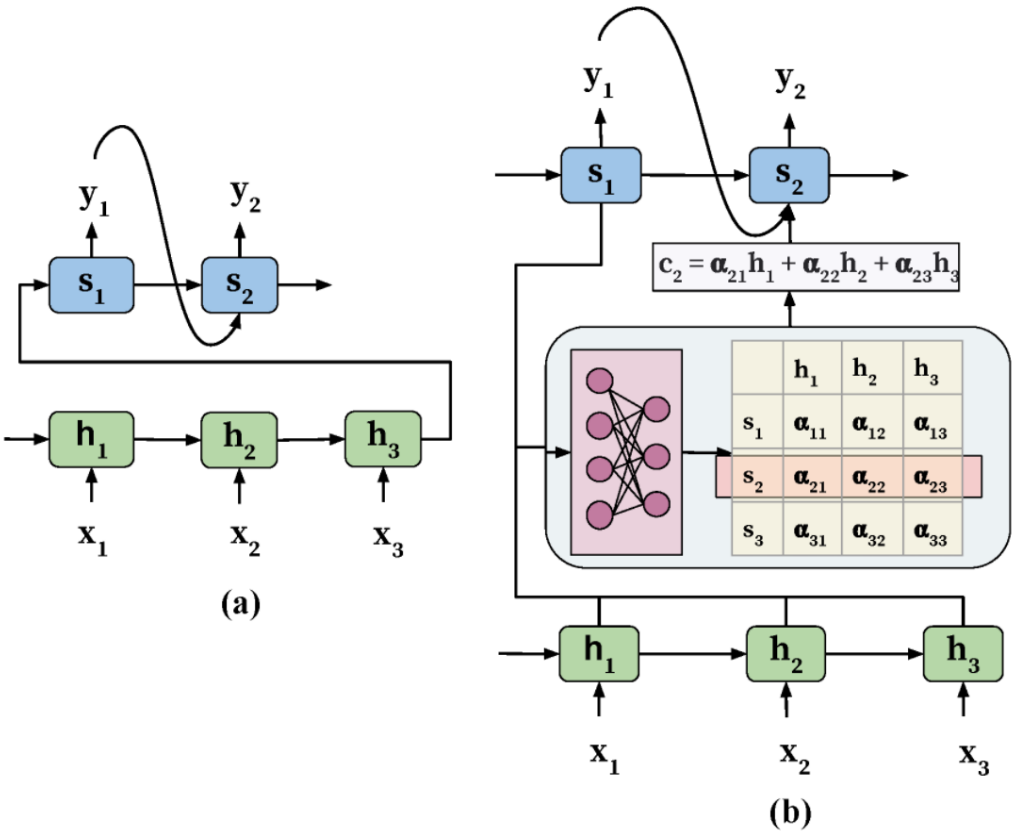

Figure 2.2: (a) Traditional Encoder-decoder structure (b) Encoder-decoder structure of the model with the attention mechanism

RNN, and then use backpropagation to train the entire model end-to-end to classify and predict whether the patient will suffer from sepsis (Futoma et al., 2017). The other established a TCN model embedded in the framework of a multi-task Gaussian process(MTGP) adaptor to make it immediately applicable to time series data with uneven spacing (Moor, Horn, Rieck, Roqueiro & Borgwardt, 2019). The model is helpful for early sepsis prediction. Compared with MPG-AttTCN, MPG-TCN lacks the interpretability required by the medical field, so an attention-based neural network is introduced. Note that it is often used for natural language processing (Yang et al., 2016; Yu et al., 2018), and it is also used for image analysis (Mnih, Heess & Graves, n.d.; Schlemper et al., 2019).

There are several studies that integrate attention processes to increase time series forecasting model performance. To increase the performance of time series prediction, a model is built on a two-stage attention RNN. At any one moment, the encoding of such a model comprises information on both the current time point and all prior time points observed by the recurrent model (Qin et al., 2017). For the diagnosis of myotonic dystrophy, a model combining attention and TCN is also used. The model is based on the time-series diagnosis of grip strength, but it only focuses on time points and does not focus on different characteristics (Lin, Xu, Wu, Richardson & Bernal, n.d.). A study on the prediction of time series, TCN is created by combining causal convolution, dilated convolution, residual connection, and fully connected access methods. Handwriting recognition, audio synthesis, and natural language processing are just a few of the jobs that TCN has excelled in (Y. Liu, Dong, Wang & Han, 2019).

ICU sepsis prediction is an important and timely issue, timely prediction of sepsis can effectively improve the survival rate of patients and it is also an active research area. The above introduces some methods of sepsis prediction, which can be benchmarked against each other. I will take this idea in this work, develop 1 machine learning models, develop 2 deep learning models, and compare their performance.

2.3.2 Research on Other Artificial Intelligence Method to Predict Sepsis

In 2021, scholars such as Goh developed a new artificial intelligence SERA algorithm (Goh et al., 2021). The data of this algorithm is a combination of structured data and unstructured clinical annotation data. They established a new artificial intelligence model called SERA to predict and diagnose sepsis. The model uses natural language processing (NLP) analysis of doctors' clinical notes, and then combines them with structured EMR data to use rich data resources to improve the performance of the model for predicting the risk of patients suffering from sepsis. The study included 327 patients with confirmed sepsis, of which 240 patients were used for model training and verification, and the remaining 87 were used as test samples without verification samples. In addition, there is another problem. Since only 6.15% of the 327 patients suffer from sepsis, the researchers used the oversampling technique (SMOTE) that generates minority samples to achieve a 1:1 balance of data, which is worth learning from. The SERA model designed by the researchers implements two algorithms. The diagnosis algorithm first determines whether the patient suffers from sepsis during the outpatient clinic. If not, the early warning algorithm will determine whether the patient suffers from sepsis at different time periods, including 4, 6 , 12, 24, 48 hours. Testing the algorithm through independent clinical annotations, the model achieved a high prediction accuracy 12 hours before the onset of sepsis, and its AUC was 0.94.

2.4 MIMIC-III Database

MIMIC-III (Medical Information Mark for Intensive Care) is an intensive care data set released (Goh et al., 2021). It was abbreviated as MIMIC-II at the beginning of its establishment, and upgraded to MIMIC-III in 2016. It contains a total of more

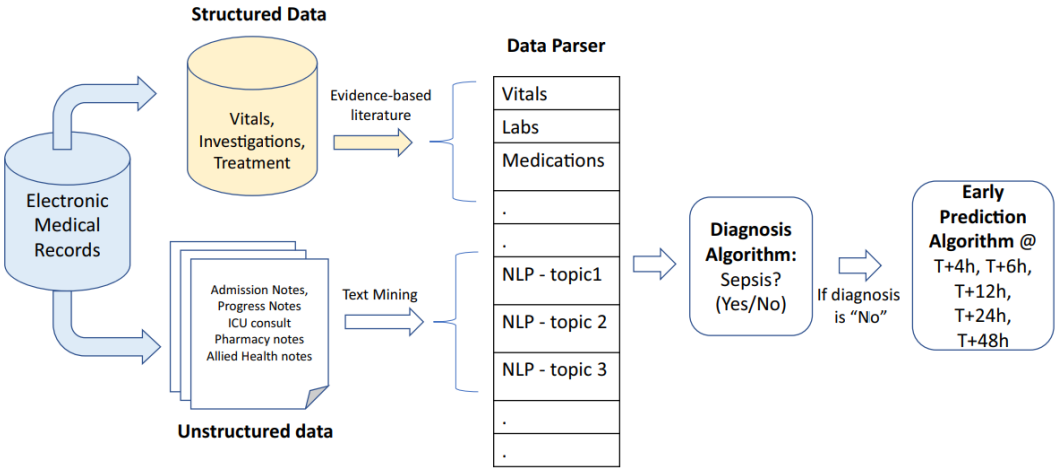

Figure 2.3: Flow chart of SERA algorithm
Source:(Goh et al., 2021)

than 50,000 patient hospitalization data and nearly 60,000 ICU hospitalization records who were hospitalized in the ICU from June 2001 to October 2012. It contains two types of basic data: one type is clinical data extracted from EHR, including patient demographic information, diagnostic information, laboratory test information, medical imaging information, vital signs, etc. The other is the waveform data collected by bedside monitoring equipment and related vital sign parameters and event records. It is worth noting that the database records the follow-up results of patients, which provides the possibility for us to carry out prognosis research in the future (A. E. W. Johnson et al., 2016). A large number of studies at home and abroad show that the database has now achieved a perfect combination with clinical practice. It is the world's first large-scale intensive care unit database that is free and open to the public (Figure 2.4). It has a wealth of medical data types. According to the data usage agreement, international researchers can obtain these data widely, and offer high-quality data for clinical research and data mining, and building a knowledge base (Q. Liu et al., 2019).

The MIMIC-III database is a relational database that supports SQL query, and can import data sets into large relational databases such as SQL Server, Postgres, MySQL,

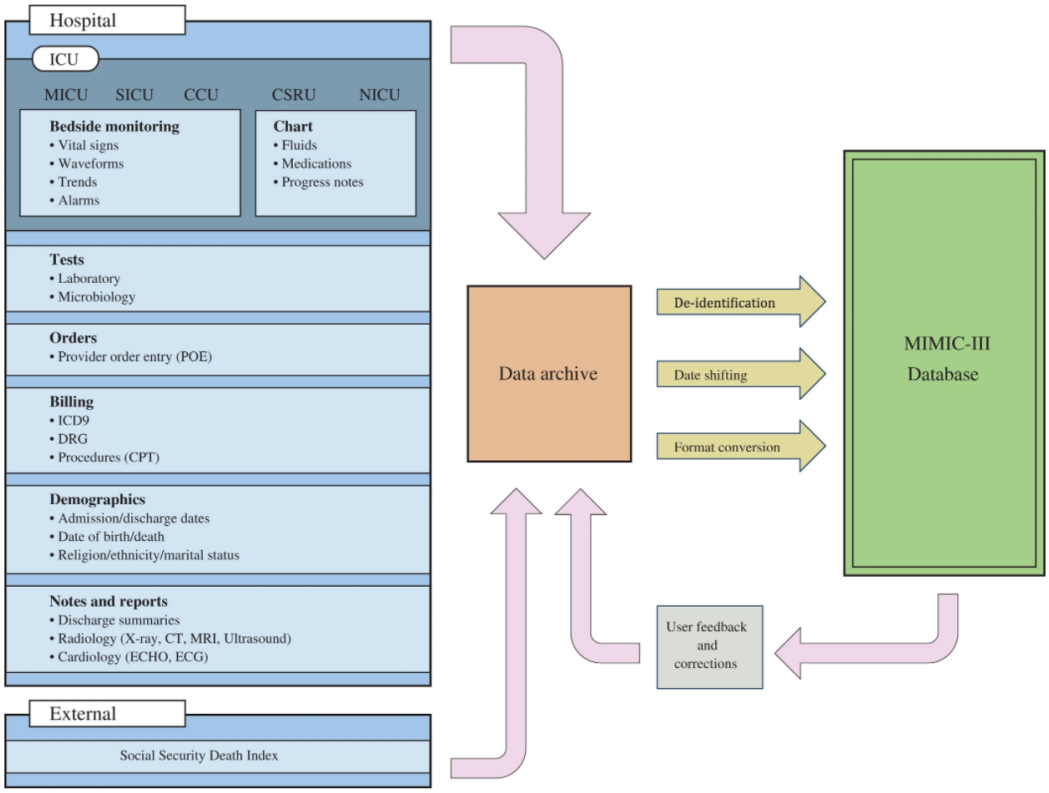

Figure 2.4: Basic structure overview of the MIMIC-III critical care database
Source: (Adapted from A. E. W. Johnson et al., 2016, Figure 1., pp. 160035–3)

and Oracle for data processing. At the same time, the table data in the data set can be exported in .csv format, and large statistical software such as Excel, Spss, Matlab can be used for data processing and statistical analysis guidance research (A. E. W. Johnson et al., 2016).

The MIMIC-III database contains 26 data tables. Except for the dictionary table, the tables are connected by subject_id, hadm_id, and icustay_id (A. Johnson, Pollard & Mark, 2015). The information table is divided into three categories, demographic information of the patient, information collected during the patient's ICU hospitalization, and information collected by the hospital recording system.

Table 2.2: Tables in MIMIC-III

	Tables Name	**Description**
†[1]	**1:** ADMISSIONS	Patient admission
†	**2:** CALLOUT	Instant information when the patient is out of the ICU [2]
†	**3:** ICUSTANYS	ICU admission information
†	**4:** PATIENTS	Patient information
†	**5:** SERVICE	Medical services that patients need to receive
†	**6:** TRANSFERS	Patient Turnover Information
†[3]	**7:** CAREGIVERS	Nursing Staff Information
†	**8:** CHARTEVENTS	Patient observation and recording data
†	**9:** DATETIMEEVENTS.	Patient operation time information
†	**10:** INPUTEVENTS_CV	Intake information recorded using CareVue
†	**11:** INPUTEVENTS_MV	iMDSoft Metavision system input data
†	**12:** NOTEEVENTS	Treatment record
†	**13:** OUTPUTEVENTS	Patient output information
†	**14:** PROCEDUREEVENTS_MV	Metavision system operating system

Continued over page

[1] demographic information of the patient(1-6).

[2] "Admission time." the admission time is more important for us to collect patient information in a specific time window

[3] information collected during the patient's ICU hospitalization(7-14).

Table 2.2: Tables in... *(continued)*

	Tables Name	Description
†[4]	**15:** CPTEVENTS	Patient operation record
†	**16:** DIAGNOSES_ICD	Patient diagnosis ICD9 code
†	**17:** DRGCODES	Patient diagnosis category group [5]
†	**18:** LABEVENTS	Patient test items
†	**19:** MICROBIOLOGYEVENTS	Results of microbial pathogen detection
†	**20:** PRESCRIPTIONS	Patient medication records
†	**21:** PROCEDURES_ICD	Patient operation record ICD9
† [6]	**22:** D_CPT	Operation record code index
†	**23:** D_ICD_DIAGNOSES	Diagnostic code index
†	**24:** D_ICD_PROCEDURES	Surgical operation code index
†	**25:** D_ITEMS	Record item code index
†	**26:** D_LABITEMS	Test item code index

Most of the sepsis prediction literature we mentioned above are constructed and tested using the MIMIC database (Desautels et al., 2016; Lin et al., n.d.)

2.5 Using AI Methods

2.5.1 Decision Tree

Decision Tree originated from the Concept Learning System and was proposed by Hunt et al. in the 1960s (Hunt, Marin & Stone, 1966). The decision tree algorithm can infer classification rules from a set of unordered and unruly examples, which is equivalent to a Boolean function.

[4]information collected by the hospital recording system(15-21).
[5]Record the patient's diagnosis category and diagnosis code.
[6]Dictionary information (22-26).

Decision tree structure

Decision tree is a tree structure, including root nodes, internal nodes and leaf nodes. Each internal node represents a test on the property, each branch represents a test outcome, and the class label (Zhou & Chen, 2002) is stored in the leaf node. After constructing the decision tree model, the model can be used to classify new samples. Assuming that a tuple X with an unknown class label is given, and the category of the tuple is tested, then the tuple X can be matched with a path from the root node to the leaf node, and the leaf node stores the class prediction of X .

Decision tree attribute selection

The attribute selection of the decision tree needs to be determined according to the attribute measurement value. Attribute measurement is a heuristic method for judging how to split the index and dividing the training tuple D of a given class label into separate classes. Ideally, each partition is pure. The literature shows that there are three main measurement criteria: information gain, information gain rate and Gini index. (Olanow, Watts & Koller, 2001; Kotsiantis, 2013). The commonly used measurement criterion is information gain, and the information gain calculation method of the attribute A is as follows:

$$Gain(A) = Info(D) - Info_A(D)$$

That is, the gain of attribute A is the difference between the expectation of D and the expectation after dividing the tuple of D by attribute A.

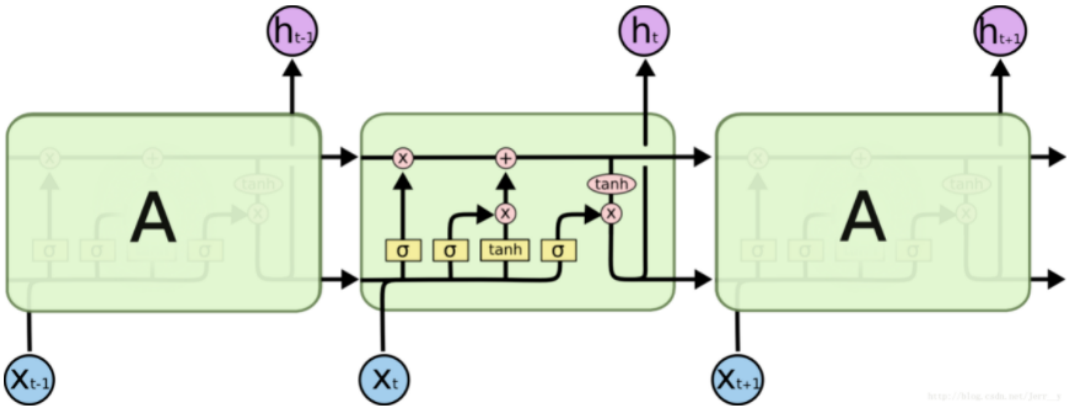

Figure 2.5: LSTM internal structure
(Elsaraiti & Merabet, 2021)

2.5.2 LSTM Neural Network

Theoretical of LSTM

The LSTM neural network is a form of deep machine learning neural network that is based on RNN (Hua et al., 2018). RNN is a network structure specially used to process sequence data, and its connection mode can make it convenient to extract time information of sequence data (Sherstinsky, 2020). There is a time-domain connection between a series of data input by RNN, which takes into account the "memory" information. We know that the training of the RNN network is achieved through an algorithm called Back Propagation Through Time (BPTT) (Liao et al., 2019). When the sequence becomes longer, the problem of gradient disappearance will occur during the back propagation process. LSTM is put forward by researchers to solve this problem. The structure of LSTM is shown in Figure 2.5.

The current input x_t and the prior hidden state h_{t-1} are spliced to get $[x_t, h_{t-1}]$, as shown in the diagram, and the spliced vector is subjected to respective operations to

obtain the four-part output as follows:

$$f_t = \sigma(W_f \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_f)$$

$$i_t = \sigma(W_i \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_C)$$

$$o_t = \sigma(W_o \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_o)$$

Among them, the activation function σ is the sigmoid function, $\tilde{C}_t$ is obtained from the splicing vector through the tanh function, which represents the unit state update value. f_t, i_t, o_t correspond to the three gated states, also called gated switches, f_t is called the forget gate, through which we can know which elements of C_{t-1} will be reserved for calculating C_t. i_t is called the input gate, which is used to control which elements of $\tilde{C}_t$ will be reserved for calculating C_t. o_t is called the output gate, which together with C_t determines the final output h_t.

Further calculations are as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$h_t = o_t \odot tanh(C_t)$$

$$y_t = \sigma(w' h_t)$$

$\odot$ is the multiplication of the corresponding elements of the two matrices.

Through these gated switches, LSTM effectively solves the long-term dependency problem of RNN networks (Liao et al., 2019). And research has shown that LSTM performs better than RNN in many sequence modeling tasks (Hochreiter & Schmidhuber, 1997).

Advantages and Disadantages of LSTM model

The birth of LSTM is due to the fact that once an RNN is turned into a super long conventional neural network, the error will progressively reduce when employing BP backpropagation, but because the expansion is too long, the error must be assigned to each layer and each Neuron. This will cause the error to vanish once it has computed half of the gradient, and the vanishing of the gradient will make the training weight update change very tiny, causing the entire training process to fail to escape the local optimum solution (Hochreiter & Schmidhuber, 1997).

Specifically, it is modified based on the RNN, and each neuron in each layer is set with three gates, which are input gate, output gate and forget gate. It is possible to selectively forget and partially or completely accept according to the feedback weight correction number, so that every neuron will not be modified, so that the gradient will not disappear multiple times, thus the weights of the previous layers can also be obtained modify, and, the error function drops quicker with the gradient. For the feedback error attribution of the RNN neural network, LSTM provides a more flexible learning procedure so that it does not rapidly reach the local optimal solution during the gradient descent phase. However, in theory, LSTM can't entirely rule out the possibility of local optimum solutions (Xie et al., 2019).

2.5.3 Temporal Convolutional Network

Since RNN was first developed, after continuous evolution and iteration, the problem that was difficult to implement backpropagation was solved by using an improved model of LSTM, and the GRU developed after LSTM was simplified can basically maintain a certain degree of goodness. Therefore, RNNs have been the king of solving sequence problems for a long time (Nan, Trăscău, Florea & Iacob, 2021). However,

in 2016, some researchers pioneered the first time convolutional network for video-based action segmentation (Lea, Vidal, Reiter & Hager, 2016). Although convolutional neural networks (CNNs) are usually associated with image classification tasks, with appropriate modifications, they have proven to be valuable tools for sequence modeling and prediction. The thinking mode of recurrent neural networks is outdated. When modeling sequence data, convolutional networks (Bai, Zico Kolter & Koltun.v., 2018) should be the first consideration. Research can show that convolutional networks can achieve better performance than RNNs in the prediction of many time series tasks, while avoiding common defects of recursive models, such as the problem of vanishing gradients. In addition, using a convolutional network instead of a recursive network can improve performance because it allows the output to be calculated in parallel.

TCN is the abbreviation of Temporal Convolutional Network, which consists of an expanded, causal one-dimensional convolutional layer with the same input and output length. Before introducing temporal convolution, first briefly introduce the convolutional network. The core of the convolutional neural network(CNN) is the convolution operation. the convolution operation refers to the inner product operation of data and a set of fixed weight filter matrices. Convolutional neural networks are commonly used in the image field, and with the need for parallel computing and improvement of RNN networks in the field of time series prediction, the convolutional network has also been transformed into a time series convolutional network suitable for the needs of time series prediction (Gu et al., 2017).

One-dimensional convolution

The significance of the one-dimensional convolution used by the TCN network is to realize the element-level prediction of the sequence (J. Liu et al., 2021). In addition, the higher the level of convolution operation, the more sensitive to feature changes. Therefore, TCN uses one-dimensional convolution instead of full connection, so that

the output and input dimensions are consistent, and the end-to-end prediction effect is realized. It not only helps to feel the information of the entire input sequence, but also helps to build long-term memory.

Causal Convolutions

Causal convolution can be visually represented in Figure 2.6. That is, for the value at time t of the previous layer, it only depends on the value at time t and before in the next layer. TCN guarantees that when calculating the time step t, only the information of the time step $(t-1)$ and the previous time step $(t-1, t-2, t-3, ...)$ will be used. Causal convolution differs from standard convolutional neural networks in that it cannot see future data. It is a one-way structure rather than a two-way structure. That is to say, only the first reason may result in the second. It's termed causal convolution since it's a model with a strong temporal restriction. In form, it is similar to "cutting off" the second half of one-dimensional convolution. Therefore, it is convenient to process data related to timing.

Formally, for a one-dimensional input sequence $X = (x_1, x_2, ..., x_T)$ and convolution kernel $F = (f_1, f_2, ..., f_K)$, the causal convolution at x_t is:

$$(F * X)(x_t) = \sum_{i=1}^{K} f_k x_{t-K+k}$$

Dilated Causal Convolution

The modeling length of time is restricted by the size of the convolution kernel in pure causal convolution, just as it is in standard convolutional neural networks. We need to stack several layers linearly if we want to capture longer dependencies. In order to address this issue, the researchers proposed dilated convolution. Dilated Causal Convolution is also called hole convolution or expansion convolution, which is the

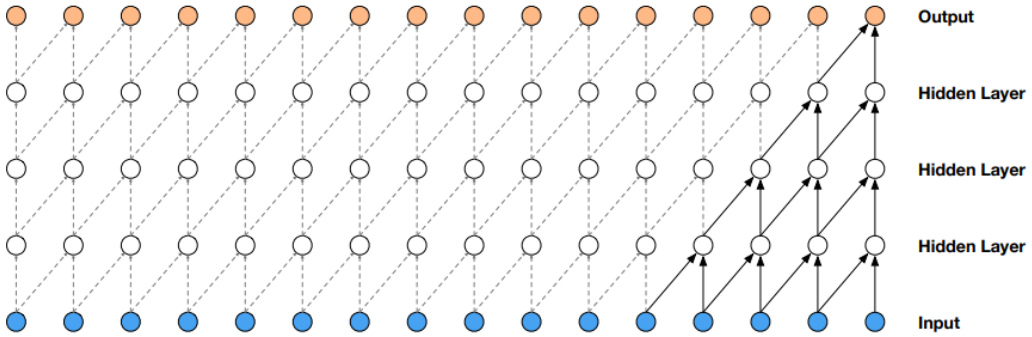

Figure 2.6: causal convolution layers *(Oord et al., 2016)*

operation of the dilation parameter.

TCN expands the receptive field by skipping some existing pixels. The number of skipped pixels is (dilation-1), which can also be understood as how many zeros are inserted between every two convolution kernels. This feature ensures that the network can observe a larger sequence length while the amount of calculation is basically unchanged.

Figure 2.7 can more intuitively feel the network architecture of TCN, this is a five-layer neural network, in which the convolution kernel is 4, the first layer is the input layer, and the dilation of the second hidden layer is 1, which is also a conventional one-dimensional causality Convolution operation. The dilation size of the latter level is doubled in turn, that is, a "jump mechanism" is added. By skipping more pixels, it is ensured that the neurons in the upper layer have a larger receptive field. Here, the last neuron in the uppermost layer can observe a total of 16 input data, and can observe all the features of the input data, while the conventional one-dimensional causal convolution can only observe 8 input data under the same circumstances.

Formally, for a one-dimensional input sequence $X = (x_1, x_2, ..., x_T)$ and convolution kernel $F = (f_1, f_2, ..., f_K)$, the dilated convolution with dilation factor d at x_t is:

$$(F *_d X)(x_t) = x_t) = \sum_{i=1}^{K} f_k x_{t-(K-k)d}$$

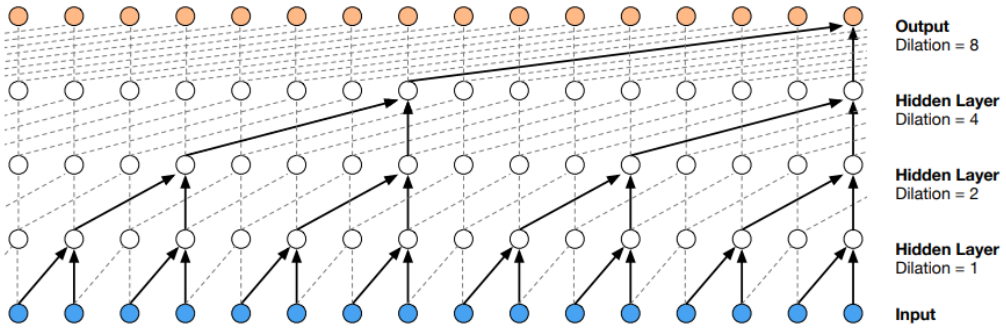

Figure 2.7: dilated causal convolution layers*(Oord et al., 2016)*

In the above formula, d is the dilation coefficient, k is the size of the convolution kernel, and $(K-1)d+1$ is the receptive field, which indicates which unit of the upper layer is used when calculating the next layer of neurons.

In addition, it can also be seen that each layer of the network has the same dimensions, which is the structural feature of one-dimensional full convolution.

Residual Connections

The residual network solves the training problem of the deep network through its unique connection method, and greatly increases the number of layers of the network. In order to ensure the stability of deep-level TCN network training, the TCN network introduces a residual network connection method. That is, in the case of high-dimensional input, TCN uses a residual module to deepen the convolutional network, where the residual block is connected as shown in Figure 3.4, and the output of the residual block combines the input information and the output information of the input convolution operation.

In summary, TCN's typical convolution layer contains one-dimensional full convolution, causal convolution, and dilated convolution, and every two of these convolutional layers, as well as identity mapping, is wrapped into a residual block, as illustrated in Figure 2.9. Finally, a number of such residual blocks can be stacked into a deep TCN

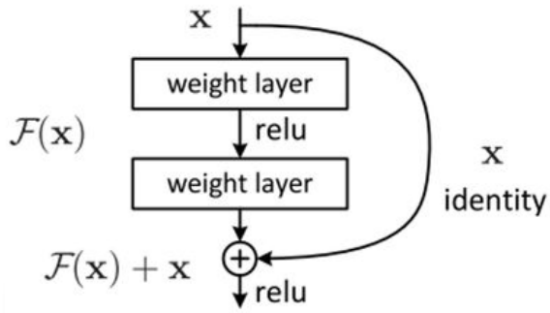

Figure 2.8: Residual Connections layers

network. Here, the number of residual blocks in the TCN network is N. From top to bottom, the expansion coefficient of the expansion causal convolution in each residual module The dilation increases exponentially, dilation = 2^{i-1}, where$i = 1, 2, ..., N - 1$.

2.5.4 Model Evaluation

Model evaluation is used to evaluate the performance of the model. The training error, also known as the empirical error, is the mistake that the model makes on the training set, whereas the generalization error is the error that the model makes on the new sample (*Model Evaluation Metrics in Machine Learning*, n.d.).

Our goal is to use 12-hour observational health data to predict whether sepsis will occur in following 6 hours, which is obviously a binary classification problem. Regardless of the establishment of a machine learning model or a deep learning model, the final evaluation of the performance of the model can be regarded as the evaluation of the classification model. Of course, our ultimate goal is to build a learner that solves the problem with a small generalization error. However, in practical applications, the new sample is unknown, so the training error can only be made as small as possible. Therefore, in order to obtain a model with a small generalization error, when building a machine model, the data set is usually split into independent training data sets, verification data sets, and test data sets. During the training process, the validation data set is used to evaluate the model and update the hyperparameters. After training, use

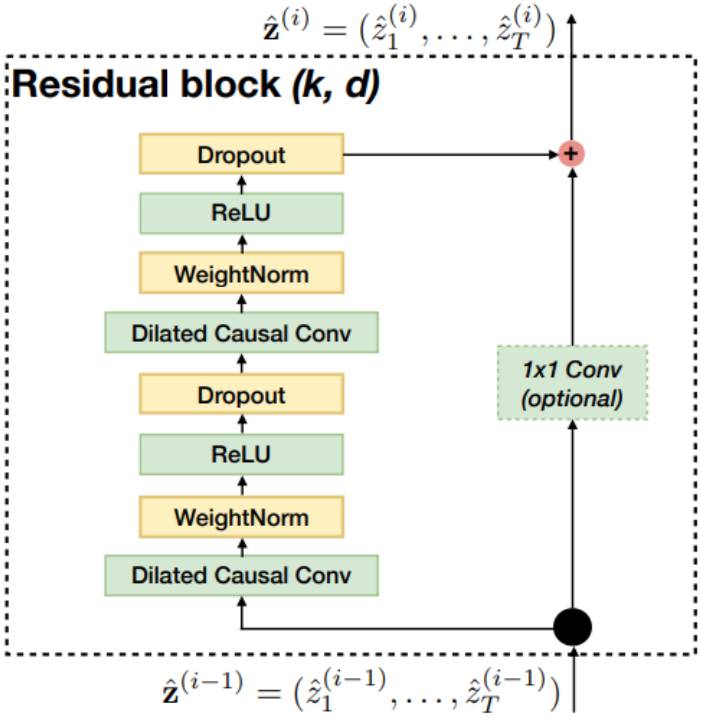

Figure 2.9: Residual block*(Bai, Zico Kolter & Koltun.v., 2018)*

the test data set to evaluate the performance of the final trained model.

Model multiple evaluation indicators

Here a table 2.3 is used to summarize the relevant evaluation indicators of the classification model.

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

In classification issues, accuracy is the simplest and most straightforward assessment index, yet it has apparent flaws. For example, when the samples are unbalanced, for example, the ratio of positive and negative samples is $1:9$, and the classifier predicts all samples as negative samples and can obtain an accuracy of 90%.

Table 2.3: Classification model evaluation index

evaluation index	**Description**
Confusion Matrix	**TP:** True Positive **TN:** True Negative **FP:** False Positive **FN:** False Negative
Accuracy	The percentage of samples that were properly categorized out of the total number of samples.
Precision	The ratio of properly categorized positive samples to the total number of positive samples determined by the classifier.
Recall	The number of accurately categorized positive samples divided by the number of actual positive samples.
F1 Score	Precision and recall rates are averaged in a harmonic way.
ROC Curve	receiver operating characteristic curve.
P-R Curve	Precision-Recall Curve.

Precision

$$Precision = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{TP}{TP + FN}$$

F1 Score

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

ROC Curve The true positive rate (TPR) is shown by the vertical axis of the ROC curve, while the false positive rate is represented by the horizontal axis (FPR).

$$TPR = \frac{TP}{TP + FN} = Recall$$

$$FPR = \frac{FP}{FP + TN}$$

TPR is the recall rate, in fact, FPR is the recall rate of the negative sample angle, that is, the false call rate.

The magnitude of the area under the ROC curve is referred to as the AUC, and this number can quantitatively reflect the model's performance as evaluated by the ROC curve. The AUC value is calculated by integrating along the horizontal ROC axis. The AUC value is often between 0.5 and 1 since the ROC curve is frequently above the line $y = x$. The larger the AUC, the more likely the classifier is to rank the real positive samples first, and the better the classification performance. TThe ROC curve should be as near to the upper left as possible.

P-R(Precision-Recall) Curve The recall rate is on the horizontal axis of the P-R curve, while the precision rate is on the vertical. For the classification model, Under a particular threshold, a point on the P-R curve signifies that the model assesses the result greater than the threshold as a positive sample, and the result smaller than the threshold as a negative sample, and returns the corresponding result at this time Recall rate and precision rate. The PR curve should be as near to the upper right as possible.

Classification model evaluation Scikit-learn function We will use the model evaluation index function provided in scikit-learn (*Scikit-Learn - Model Evaluation & Scoring Metrics*, n.d.). The classification model evaluation Scikit-learn function is shown in table 2.4.

Table 2.4: Scikit-learn function

evaluation index	**Scikit-learn function**
Confusion Matrix	from sklearn.metrics import confusion_matrix
Precision	from sklearn.metrics import precision_score
Recall	from sklearn.metrics import recall_score
F1	from sklearns.metrics import f1_score
ROC	from sklearn.metrics import roc
AUC	from sklearn.metrics import auc

Chapter 3

Methodology

This thesis plan predicts whether ICU patients will suffer from sepsis at a certain time in the future. One of the first, we have to decide what research method to use. Therefore, this chapter first considers the research worldview and philosophical hypothesis methods (Hu & Chang, 2017), and the worldview and research methods guide the further specific research process. This thesis first briefly describes the four philosophical worldview theories, and chooses the worldview that suits our research questions. Then three research methods are described, and the research method suitable for our problem is selected. Each method has its own philosophical worldview, which also influences research methods.

This chapter is composed of seven parts. The first part is the research involved and is the basis for guiding the entire experiment. The philosophical worldview and research methods suitable for the research problem have been selected. The second part introduces the whole experiment process. Then elaborated on the process of data collection and processing. Use two parts to introduce two code bases, one is the Spesis-mimic code base, and the other is the MIMIC-III code base. Then introduce how to install postgreSQL, how to import MIMIC-III database. Then it will introduce how to extract sepsis-related data from postgreSQL, how to label pivoted vital data, and split it

into three data sets for model training, validation and testing according to 7:1:2. Finally, the corresponding feature sequence data set is constructed.

3.1 Research Design

Research Design is a research method designed according to the research topic. This research method is how to collect data, what kind of data to collect, how to analyze the data, and finally answer our research questions (Tawfik et al., 2019). Research method is the foundation. Each method has its own philosophical world view. So this module includes philosophical worldview selection and research method design.

3.1.1 Philosophical Worldview

The philosophical world view is a methodology, a set of basic beliefs that guide actions (Kingma & Ba, 2015). Creswell mentioned four worldviews in his article, each with its own characteristics.

- **Postpositivist:** the purpose of postpositivist research is to use a series of meticulous and rigorous methods to "falsify" the inaccurate appearances and gradually approach objective reality.
- **Constructive:** Constructivism believes that learning is the process of generating meaning and constructing understanding based on the original knowledge and experience, and this process is often completed in social and cultural interaction.
- **Transformative:** the most important thing to transformative world view is to emphasize development.
- **Pragmatic:** the pragmatic is the discovery of natural laws or constant relationships in facts through observation and experience. This worldview is mainly about experimenting under controlled conditions, then understanding and constructing.

Obviously, the worldview suitable for the study of this thesis is the pragmatic worldview

3.1.2 Research Approach

Research Approach includes three categories (Cohen, Manion & Morrison, 2000):

- **Qualitative Research:** the purpose of research is to understand a phenomenon and a problem. It includes Grounded Theory, Phenomenological Research, Action Research and Case Study.
- **Quantitative Research:** understand the relationship between various variables to test a theory or model. There are many quantitative analysis tools, such as SPSS, R, SAS, etc. Statistical methods are usually used for data analysis.
- **Mixed Research Approach:** it is a comprehensive method that combines quantitative and qualitative methods.

In this thesis, our goal is to use 12-hour observational data to predict the risk of sepsis in the next 6 hours, so as to prevent or make the best clinical decision in advance and improve the survival rate of patients. These observational data include vital signs data, such as heart rate, body temperature, blood pressure, etc., as well as basic individual data (age, gender, etc.). In the future, we may include experimental data, but these data are all numerical or categorical. So the most suitable method is Qualitative Research.

3.2 Experimental Process

The goal of this thesis is to Use 12-hours observational health data to predict whether sepsis will occur in following 6 hours. Therefore, the entire experiment process is to

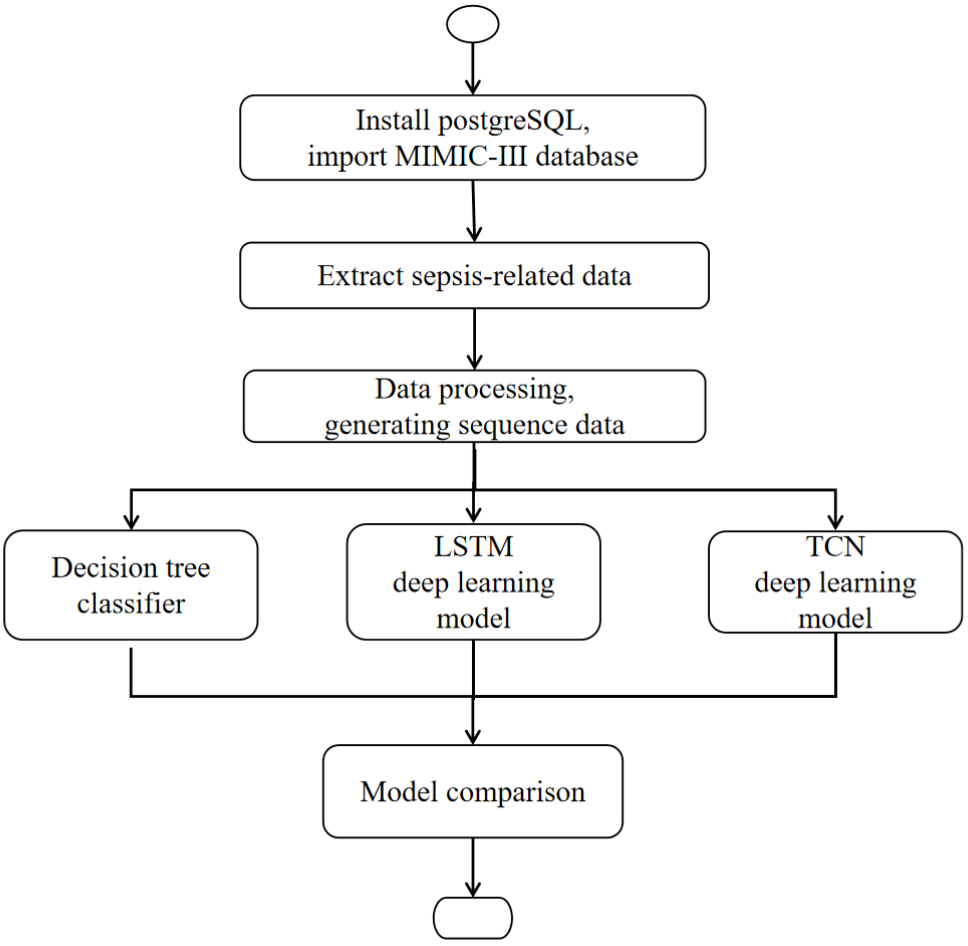

Figure 3.1: Thesis research experiment process

prepare data, model, and evaluate the model performance. The more detailed process is shown in Figure 3.1.

First of all, we will implement it based on MIMIC-III, Because MIMIC_III is clinically real data, covering a wide range, the data is hidden, safe and reliable, and my associate professor Dr. Samaneh Madanian strongly recommends it. So we need to download the MIMIC-III database, then install postgreSQL, and then import the MIMIC-III database. MIMIC-III has 26 tables, and the data we need cannot be achieved directly by exporting a table. Therefore, drawing on the research of previous scholars, the relevant codes in the two code bases of spesis3-mimic and mimic-iii are used to extract data related to sepsis prediction. Then the raw data is processed, including labeled data, which follows the new definition of spesis-3, and the data is split into training set, validation set and test set, and finally the three data sets are sequenced. This is a brief description of the data preparation process.

Then use sequence data modeling, here is a machine learning model built, it is a decision tree classifier, two deep learning models, they are the LSTM model and the TCN model. The training set is used to train the model, the validation set is used to select hyper-parameters or the optimal model, and the test set is used to test the model.

Finally, the three models were evaluated and compared, and the results were discussed in depth.

3.3 Spesis3-mimic Codebase

This code base compares the sepsis recognition algorithms in five different Mimic-iii databases, including the Sepsis-3 standard (A. Johnson, 2021). The size and severity of diseases of all five groups are measured by hospital mortality. Figure 3.2 best summarizes the results. The comparison results show that the Sepsis-3 standard based on organ dysfunction has many advantages, such as time dependence and low sensitivity to

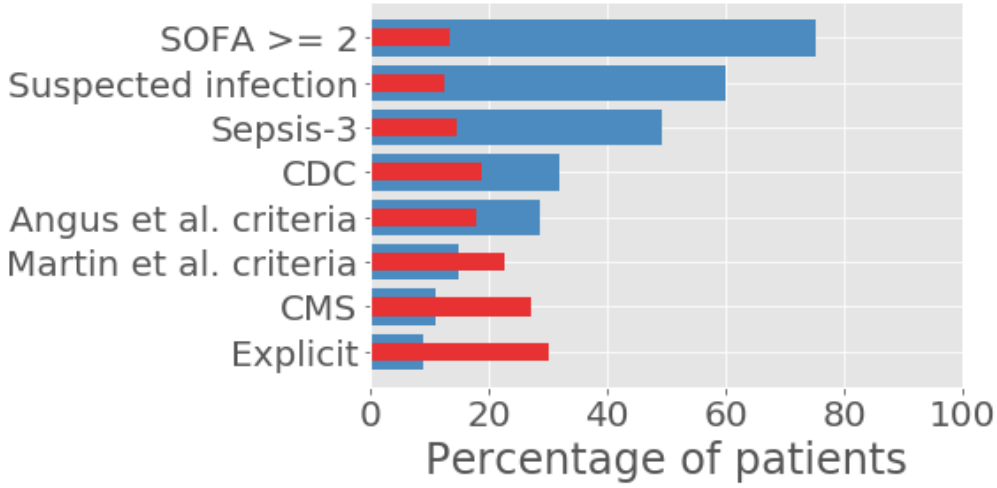

Figure 3.2: The percentage of patients with different criteria for identifying sepsis(the blue bar is the percentage of patients meeting the criteria, and the red bar is the percentage of deaths of this size)
(A. Johnson, 2021)

changes in coding practices. In the previous chapters of this project, we also mentioned using the sepsis-3 standard to identify sepsis, so we learned from the relevant code of the code base to generate relevant data. The code library interface for MIMIC-III is shown in Figure 3.3:

3.4 MIMIC-III Codebase

The MIMIC-III codebase collects and organizes clinical diagnosis and treatment information of real patients in the intensive care unit (A. E. W. Johnson et al., 2016). The database has a large sample size, comprehensive information, and a long time to track patients, but we only care about whether patients will suffer from sepsis. The coding involved in the MIMIC-III process is a challenge for data scientists (non-clinicians). The MIMIC code base provides an open source code package for analyzing patient characteristics. This code base is a useful tool for researchers to use the MIMIC database. The code library interface for MIMIC-III is shown in Figure 3.4:

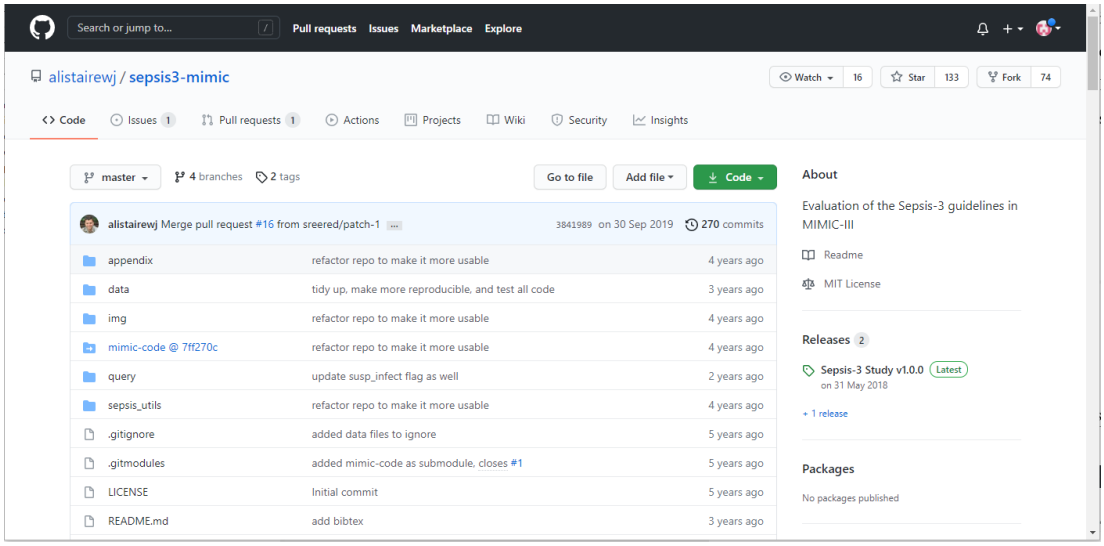

Figure 3.3: sepsis3-mimic code base interface

The code base is mainly composed of 7 parts (table 3.1), with emphasis on the content that will be used in this project.

Table 3.1: Code base content and function description

Name	Function Description
benchmark	
buildmimic	Build scripts for each relational database. This project uses related scripts to construct postgreSQL.
concepts	A useful view of the data in MIMIC-III. This project uses organ failure scores and treatment durations.
notebooks	Provides examples of how to extract and analyze data.
notebooks/aline	
tests	carry out testing.
tutorials	Explain concepts to new users.

The SQL to be used in the research process is mainly concentrated in the buildmimic and conceptual parts. Buildmimic is mainly used to build a postgres database, and the conceptual part mainly uses pivot and durations to extract sepsis-related data.

Many scholars are dedicated to the study of sepsis, but the MIMIC-IIII database does not directly indicate sepsis. Angus and Martin used management data, especially

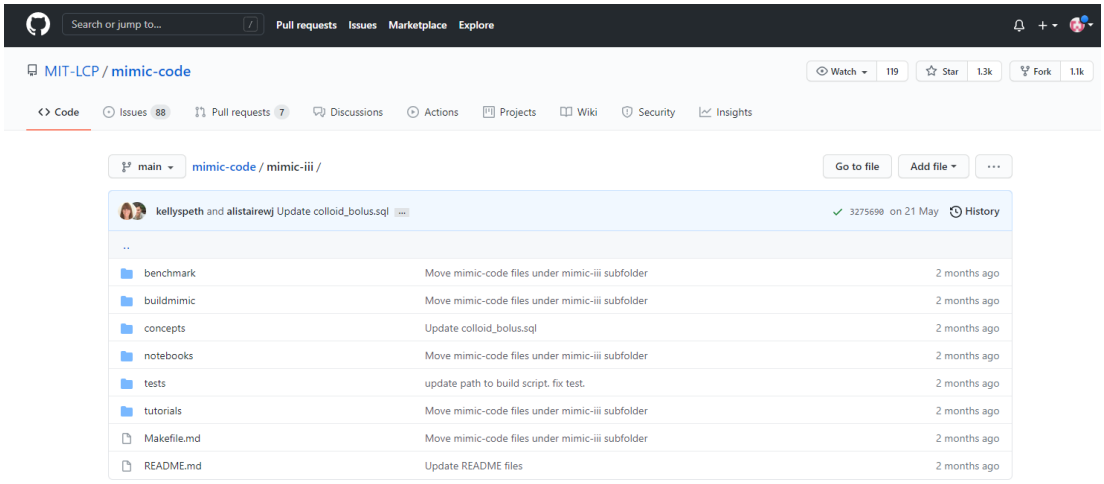

Figure 3.4: MIMIC III code base interface

the bill code obtained at the time of discharge, to regressively identify sepsis, and used a set of codes to define the algorithm for sepsis (Angus et al., 2001). This standard was verified in a follow-up study by Iwashyna (Iwashyna et al., 2014). These codes can be downloaded in sepsis in the concept section. This thesis does not use the standard proposed by Angus. Sepsis is a life-threatening organ malfunction induced by the host's uncontrolled infection response (Singer et al., 2016). Within the window of probable infection, organ dysfunction was defined as a two-point rise in the continuous organ failure assessment (SOFA). We encode the Sepsis-3 standard in the MIMIC-III data set according to the pivot and duration codes in the code base.

3.5 Data Extraction

In Sepsis-3, infected patients quantified by SOFA increase higher than 2 are defined as sepsis (Singer et al., 2016), so the SOFA scoring code is used here to obtain relevant data. We can find the database code for SOFA scoring under pivoted in the concepts of MIMIC code library. The purpose of SOFA scoring is to describe the occurrence and development of multiple organ dysfunction syndrome and evaluate the incidence,

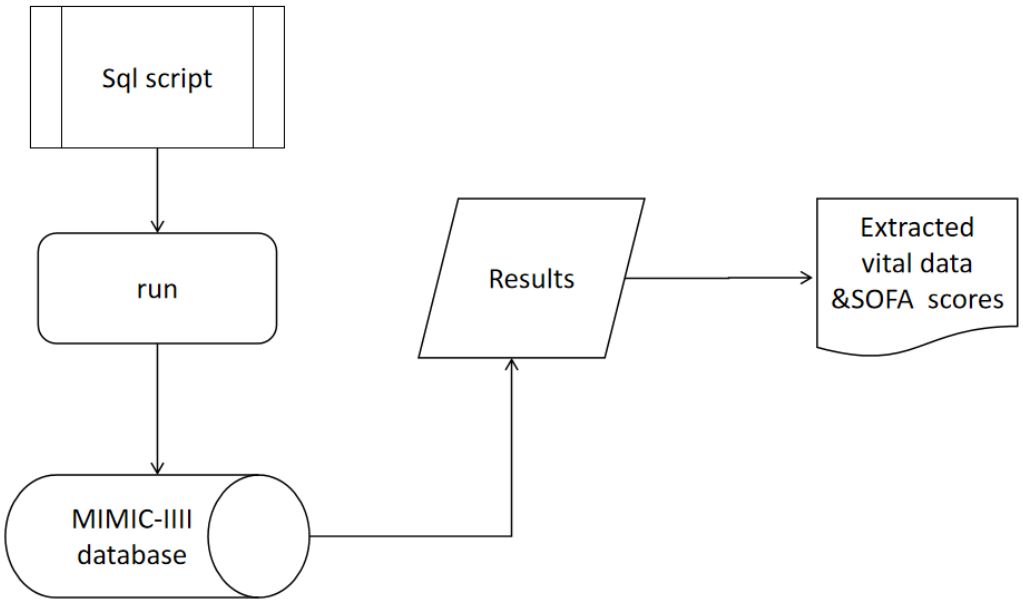

Figure 3.5: The flow chart of data extraction

quantitatively and objectively describe the severity of organ dysfunction and failure in groups of patients and even individual patients at different times (Vincent et al., 1998). The score is specific to the organ function being evaluated. It has nothing to do with patient source, type of disease, demographic characteristics and other factors, and has nothing to do with treatment measures. It can distinguish the degree of organ dysfunction or failure. It is an objective and reliable standard that medical institutions can obtain. Each medical institution can conduct daily tests by routine methods. Therefore, the use of SOFA scores in this thesis will have far-reaching practical significance.

First import the MIMIC-III data set into the PostgreSQL database. Then use the relevant sql code of the mimic code code library to obtain pivoted vital data and pivoted SOFA scores data. The flow chart of extraction data is seen Figure 3.5.

3.5.1 Install Database

Download and install PostgreSQL, download MIMIC-III data set. Then import the MIMIC-III data set into the PostgreSQL database. Follow the official instructions to import the database, and the corresponding instructions are as follows:

```
1. DROP DATABASE IF EXISTS mimic;
2. CREATE DATABASE mimic OWNER postgres;
3. \c mimic;
4. CREATE SCHEMA mimiciii;
5. set search_path to mimiciii;
6. \i D:/thesis/mimic-code-master/buildmimic/postgres/
postgres_create_tables.sql;
7. \set ON_ERROR_STOP 1;
8. \set mimic_data_dir 'D:/thesis/
mimic-iii-clinical-database-1.4';
9. \i D:/thesis/mimic-code-master/buildmimic/postgres/
postgres_load_data_7zip.sql;
```

First delete the mimic database, in case the mimic database already exists, then create your own mimic database, establish a connection to the created database, create a pattern, set the search path, then create tables, set error handling, and set the mimic data directory, and finally load data into mimic. pgAdmin 4 (*pgAdmin - PostgreSQL Tools*, n.d.) is a reliable and comprehensive database design and management software designed for PostgreSQL. After importing the data, we can see all MIMIC-III tables in pgAdmin 4, and can also query the specific data in the table and create Views, etc. (figure 3.6).

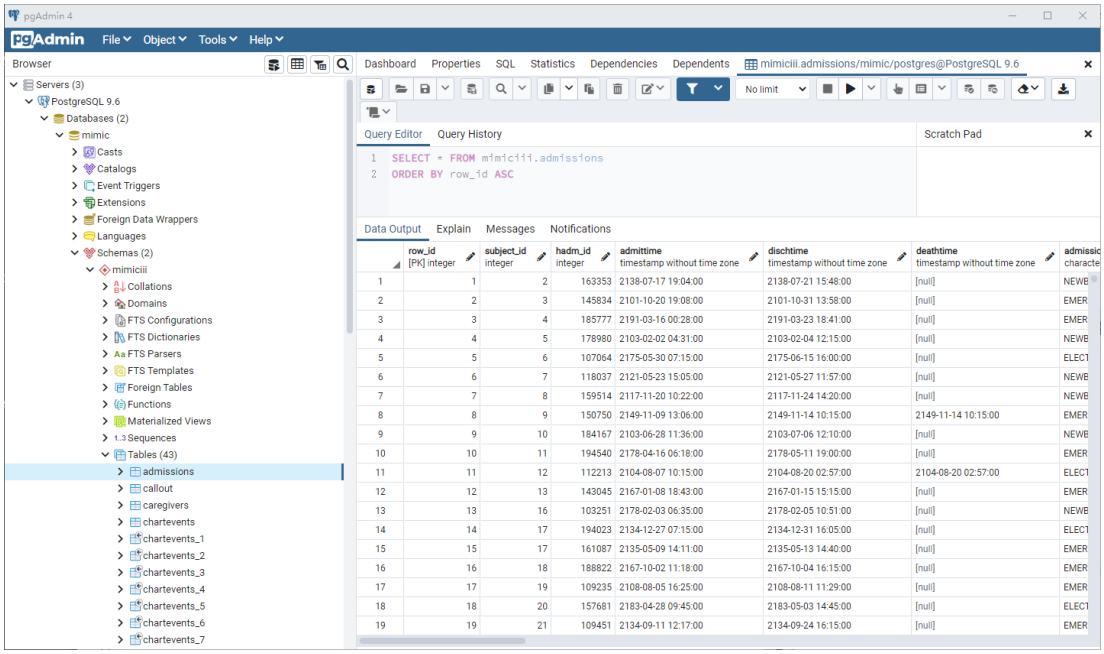

Figure 3.6: MIMIC-III database by pgAdmin 4

3.5.2 Pivoted Vital Data and Infection_time

Run *pivoted_vital.sql* in pgAdmin 4, this code creates a materialized view *pivoted_vital*. This view is the monitoring data of 8 vital signs within the first 24 hours of the patient's hospitalization. The vital signs include **Heart rate, Systolic blood pressure, diastolic blood pressure, Mean blood pressure, Respiration rate, Body temperature, Pulse oximetry and glucose**.

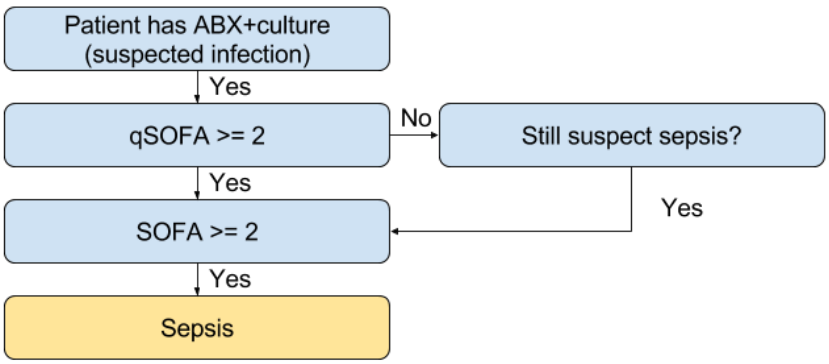

Figure 3.7: sepsis flow
Source: (A. Johnson, 2021)

We coded the Sepsis-3 standard in the MIMIC-III data set according to the (A. Johnson, 2021) code provided on GitHub. In pgAdmin 4, call the "query/make-tables.sql" script and generate the "sepsis" table on the database we need.

3.5.3 Pivoted SOFA Scores

Run $pivoted\_sofa.sql$ and its dependent view related SQL code in pgAdmin 4. After the code runs, a table $pivoted\_sofa$ is created. This table records sepsis-related organ failure assessment data. The score is calculated for each hour the patient stays in the ICU, and the window is 24 hours.

3.5.4 Generate SOFA Timeline Information

After the above two steps, there are pivoted_data view and pivoted_sofa, sepsis3 two tables, export their respective csv, which is the raw data of this thesis.

```
pivoted_vital.csv
pivoted_sofa.csv
sepsis3.csv
```

3.6 Data Preprocessing

After the above data extraction process, pivoted vital data and pivoted SOFA scores data have been obtained. Then merge the data and label the data , and then filter the data over 6 hours is the research data of our project. The entire data preprocessing process is shown in Figure 3.8.

We extract sofa score data where sofa_24hours is not null, and extract sepsis data where suspected_infection_time_poe is not null. Our processing is based on the assumption that if one or more SOFA values are missing, we assume that such patients

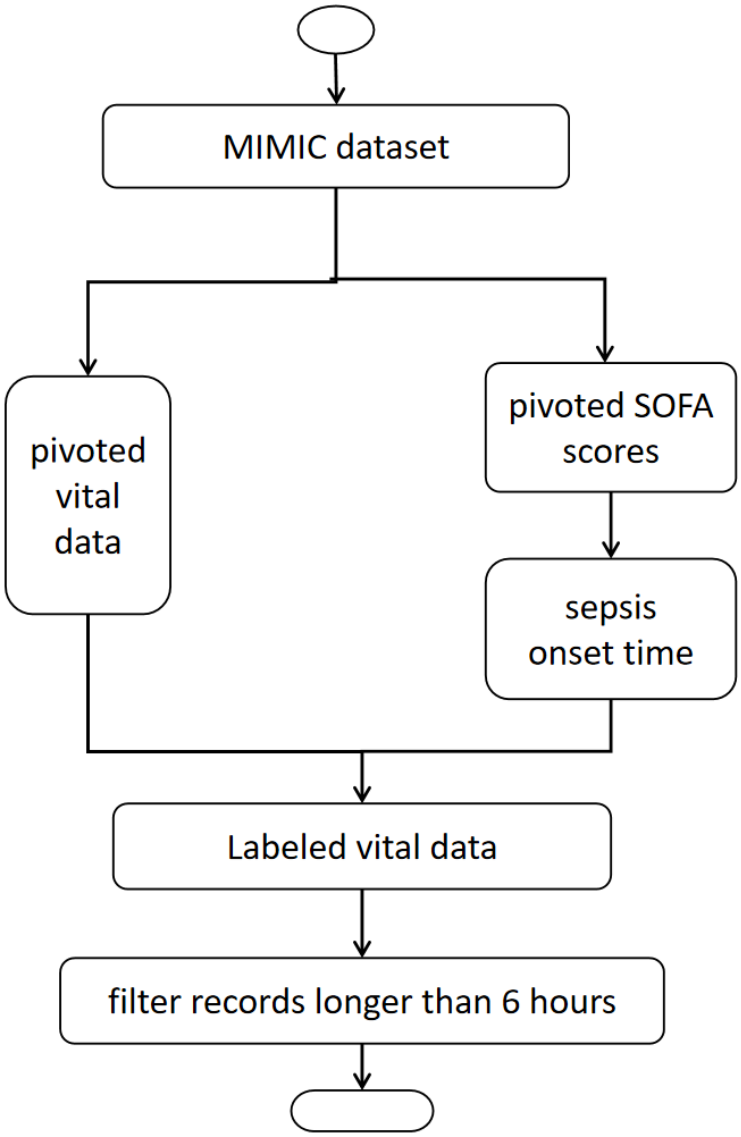

Figure 3.8: Data preprocessing process

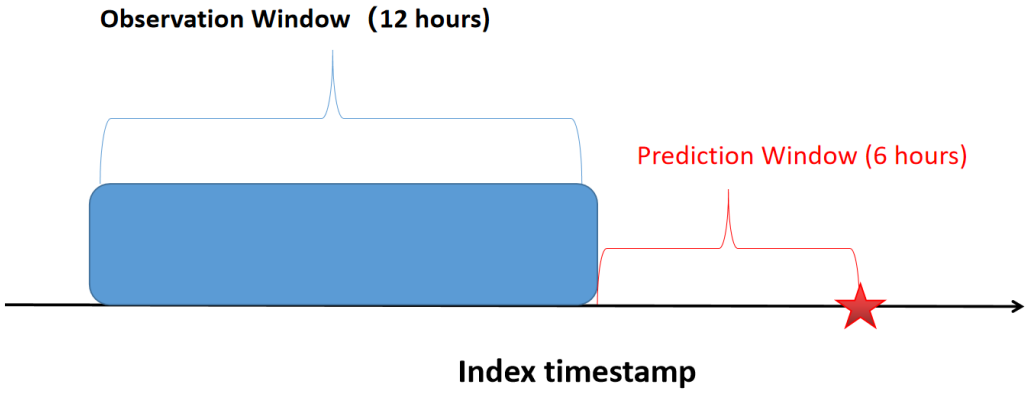

Figure 3.9: Index timestamp

are within the healthy norm and have a contribution of 0. This is a major difference from the literature (Moor et al., 2019) that does not calculate the total score. As mentioned above, the standard practice of clinicians is to assume the health value of all unmeasured variables. This is the tacit rule of the medical industry. Because doctors suspect that a certain variable exceeds the health standard, they usually measure it, so most Unmeasured variables are healthy by default. We screen patients who live in the Intensive Care Unit (ICU) and are over 14 years old with valid vital signs.

3.6.1 Get sepsis Onset Time

Retrieve the intensive care unit with sepsis and the corresponding time of onset.

3.6.2 Labeled Pivoted Vital Data

The final data processing is to prepare the labeled pivot important data for model training. In this part, three data sets are connected, namely pivoted_vital.csv, sepsis_cohort.csv and pivoted_score.csv data. In feature extraction, we extracted 8 vital signs (heart rate, body temperature...) and 2 basic variables (age, gender), and marked sepsis for each record.

Table 3.2: List the features

Vital	**Basic**	**Label**
heartrate (Heart rate)	age	sepsis_label.
sysbp(Systolic blood pressure)	gender	
diabp(Diastolic blood pressure)		
meanbp(Mean blood pressure)		
resprate(Respiratory rate)		
tempc(Body temperature)		
spo2(Pulse oximetry)		
glucose		

Through the above processing, we obtained the important data of markers including 4,647 patients admitted to the ICU with sepsis and 46,850 patients admitted to the ICU without sepsis. Then filter out the data less than 6 hours and eliminate the possibility of imbalance, so that 559 people admitted to the ICU with sepsis and a sub-sample of 559 people admitted to the ICU without sepsis are obtained.

Filter out data older than 6 hours.

```
df = final_df.groupby('icustay_id').
    filter(lambda x: len(x) > 6)
sel_ids = set(df['icustay_id'].tolist())
sel_onset_ids = set([int(x) for x in onset_time.keys().
    intersection(sel_ids)
```

3.6.3 Data Split

we divided the data into three sets: training, validation, and test in a 7:1:2 ratio. The original data was divided into three sets so that the optimal model with the highest effect accuracy and generalization capacity could be chosen. The training set's purpose is to make the model fit. The classification model is trained by adjusting the classifier's hyper-parameters. In this thesis, we use the training set to train the decision tree classifier

model, train the LSTM network model, and train the TCN deep learning network model. The purpose of validation set is to use each model to forecast validation set data and record the model's accuracy in order to determine the best model after training many models in the training set. The parameters corresponding to the model with the best effect are selected, which is used to adjust the model parameters. This thesis is mainly used for the optimization and parameter adjustment of the TCN deep learning network model. Model prediction is the function of the test set, and it is used to evaluate the best model's performance and classification abilities. The test set is a data set used to test the performance of the model. According to the error (usually the difference between the predicted output and the actual output), the quality of a model is judged. The performance comparison indicators between the three models in this thesis include Accuracy, Precision, Recall, F1-score and ROC-AUC.

3.7 Construct the Features Sequence Data

Most patients are admitted to the ICU with 24 hours of data. Our thesis has an observation window of 12 hours and a prediction window of 6 hours (see Figure 3.8). This is achieved through a trade-off between performance and window. The case patient was a patient who developed sepsis during the prediction window, while the control patient did not develop sepsis.

Through data processing, we have obtained csv files of three data sets, including training set, validation set and test set. This module needs to do the same processing on the three data sets to construct the feature sequence data for the prediction model. The observation window is 12 and the prediction window is 6 feature sequence data.

The key functions for creating a feature sequence are as follows:

Function name: `create_seqs_dataset`

Input parameters:

`path`: The directory path where the original file is located. This function is executed on the csv of the training set, validation set and test set respectively, so it is the directory path where the data set is executed at that time. This defines train_path, validation_path, tes_path in the constant variables of the program.

`observation_window=12`: Our goal is to use 12-hours observational health data to predict whether sepsis will occur in following 6 hours, so the observation window is 12. Other values can be used here, and 12 hours is the observation window for weighing performance.

`prediction_window=6`: We are predicting whether sepsis will occur in following 6 hours. We are predicting sepsis 6 hours in advance, so the prediction window is 6. In the same way, other values can be used here. According to the data performance authority, the larger the value will be obtained. The number of patients with sepsis is relatively small, so the prediction window of performance is weighed.

Return parameters:

`seqs`: records sequence data

`labels`: labels sequence data

Internal processing of the function:

The most important process is to construct features. One is to calculate how long in advance the patient can be predicted to suffer from sepsis, denoted as index_hours, and the other is to obtain the data of the construction window to construct the two output sequences.

Calculate index_hours, which can be divided into two cases: If the patient has sepsis, the number of hours of the prediction window before the onset time is the index hours for extracting the predicted sepsis. If the patient does not have sepsis, the last event is index hours, this time is greater than or equal to 6 hours of the forecast window.

After processing the three data sets, save them as the following files:

```
# Train set
sepsis.seqs.train
sepsis.labels.train
# Validation set
sepsis.seqs.validation
sepsis.labels.validation
# Test set
sepsis.seqs.test
sepsis.labels.test
```

sepsis.seqs.XXX: The independent variable squence data in the XXX data set(x_i). sepsis.labels.XXX: Response variable sequence data in the XXX data set (y)

3.7.1 Model Evaluation Criteria

This thesis predicts whether ICU patients will suffer from sepsis in the next 6 hours. This problem is a classification problem. From Chapter 2, we know that the classification algorithm indicators include Accuracy, Confusion Matrix, Precision, Recall, F1 Score and ROCAUC. These indicators will be output in the evaluation, but the final decisive indicator for judging the performance of the model is ROCAUC, and other indicators are only for reference.

Our problems are tendentious ones. Obviously, we are more inclined to identify patients who will suffer from sepsis in the future. At this time, the importance of judging that you will suffer from sepsis and actually not suffering from sepsis is different from that of judging that you will not suffer from sepsis and actually suffering from sepsis. This cannot be measured by the accuracy index alone.

The ROC curve is the relationship curve between the false positive rate (FPR) and the true positive rate (TPR). It represents costs versus benefits. Obviously, the higher the benefit and the lower the cost, the better the performance of the model. The main task of diagnosis in the hospital is to find out the people who are sick as much as possible, so the higher the TPR, the better, and try to reduce the people who are not ill and misdiagnosed as sick, that is, the lower the FPR, the better. FPR and TPR are mutually restricted. If the classifier is very sensitive, it will be easy to judge people who are not sick as being sick. If the TPR is high, the FPR will increase accordingly. It can be seen from the ROC curve that the point in the upper left corner (TPR=1, FPR=0) is a perfect classification, that is, this classifier is very good and the prediction is all right. Therefore, for our problem, ROCAUC is the most important evaluation indicator. The closer the ROC curve is to the upper left corner, the better the classifier effect, that is, the larger the area AUC under the ROC, the better the classifier classification effect.

3.7.2 Missing SOFA Processing

We know that for clinicians, the default standard practice is that all measured values are healthy values. This is because when a doctor suspects that a certain index value may deviate from the normal value, the doctor will definitely measure it, and for the unmeasured value, the doctor must be sure that there is no problem. Therefore, in this article, our treatment of the missing SOFA value means that the patient's SOFA is within the healthy standard range, so the missing SOFA value is supplemented to 0. Our labeling method conforms to standard clinical practice.

3.7.3 Unbalanced Dataset Issues Resolution

After data processing, important marker data of 4,647 patients admitted to ICU with sepsis and 46,850 patients admitted to ICU with non-sepsis were obtained here. The

data set here is obviously an unbalanced data set. The number of positive samples is much smaller than that of negative samples, and the ratio between the two is about 1:10. Then filter out the data within 6 hours. Here are 559 patients with sepsis who were admitted to the ICU. We used random under-sampling and selected 559 patients with non-sepsis who were admitted to the ICU. After this processing, the data is a balanced data.

Chapter 4

Model Implementation

This chapter mainly introduces the algorithm implementation of the decision tree, LSTM and TCN models, as well as the experimental process. Our goal is to use 12-hour observational health data to predict whether sepsis will occur in the next 6 hours. The first is to implement decision tree prediction of sepsis based on sklearn.tree.DecisionTreeClassifier. Then, based on the pytorch package of python, LSTM was implemented to predict sepsis. Finally, focus on the deep learning TCN algorithm, based on the python pytorch package to achieve TCN prediction of sepsis.

4.1 Implementation of Decision Tree

4.1.1 Decision Tree Algorithm

Most decision tree algorithms are greedy and search and traverse in a top-down, divide-and-conquer recursive manner (Safavian & Landgrebe, 1991). The general algorithm for generating decision trees is described in Algorithm 1.

2

Algorithm 1 Algorithms for Building Decision Trees

input: Training tuple D, Candidate attribute set $Attribute\_List$
output: A decision tree T

Create decision tree root node N;
if All tuples in D are of the same category C **then**
Return N is used as a leaf node and marked as category C
else if $Attribute\_List$ is empty **then**
Return N is the leaf node, and it is marked as the category C with the most categories in the sample contained in the node;
else
selects an attribute $Splitting\_criterion$ with the best classification ability from the $Attribute\_list$, and marks it as the root node N;
for each output j in $Splitting\_criterion$ **do**
According to $Splitting\_criterion$ = j, a corresponding branch is generated from the root node;
Suppose the set of array tuples in D that meets the output j is D_j;
if D_j is empty **then**
Add a leaf node and mark it as the class with the most sample categories contained in the node;
else
recursively construct the left and right subtrees, and add a node returned by $GDT(D\_j, Attribute\_list)$ to N;
end if
end for
end if
return T;

4.1.2 Decision Tree in Python sklearn library

Just as the InSight model constructed in several documents can be used as the baseline for new models, the InSight scoring model is one of some machine learning algorithms that have passed research and clinical proof-of-concept (Sheetrit et al., 2017; Dremsizov, Kellum & Angus, 2004; Mao et al., 2018; Bai, Zico Kolter & Koltun.v., 2018). We built our own machine learning model in this module, here is the decision tree classifier.

The function to build a decision tree is **sklearn.tree.DecisionTreeClassifier**, which uses the CART algorithm by default. There are three main decision tree algorithms: ID3, C4.5, and CART. The ID3 algorithm starts from the root node of the tree, always selects

the feature with the largest information gain, applies judgment conditions to this feature to establish child nodes, and proceeds recursively until the information gain is small or there is no feature. C4.5 uses the information gain ratio to select features, which is regarded as an improvement of the ID3 algorithm. But these two algorithms will lead to over-fitting problems and need to be pruned. The pruning of the decision tree is actually to remove some unnecessary classification features by optimizing the loss function, and reduce the overall complexity of the model. In the process of CART algorithm spanning tree, the classification tree adopts the principle of Gini Index minimization, and the regression tree chooses the principle of minimizing the square loss function. The CART algorithm also includes tree pruning. The CART algorithm cuts some subtrees from the bottom of the fully grown decision tree, making the model simpler. The specific implementation is through the DecisionTreeClassifier class of sklearn.tree.

4.1.3 Gini Index

The Gini index is a measure of the training tuple's impurity. The calculation method is as follows:

$$Gini(T) = 1 - \sum_{i=1}^{m} P^2(u_i)$$

In the formula, $p(u_i)$ is the probability that the tuple in T belongs to the C_i category, and is estimated by $|C_{i,T}|/|T|$ value.

Assuming that attribute A divides T into T_1 and D_2, then the Gini index of T is calculated as follows:

$$Gini_A(T) = \frac{T_1}{T} Gini(T_1) + \frac{T_2}{T} Gini(T_2)$$

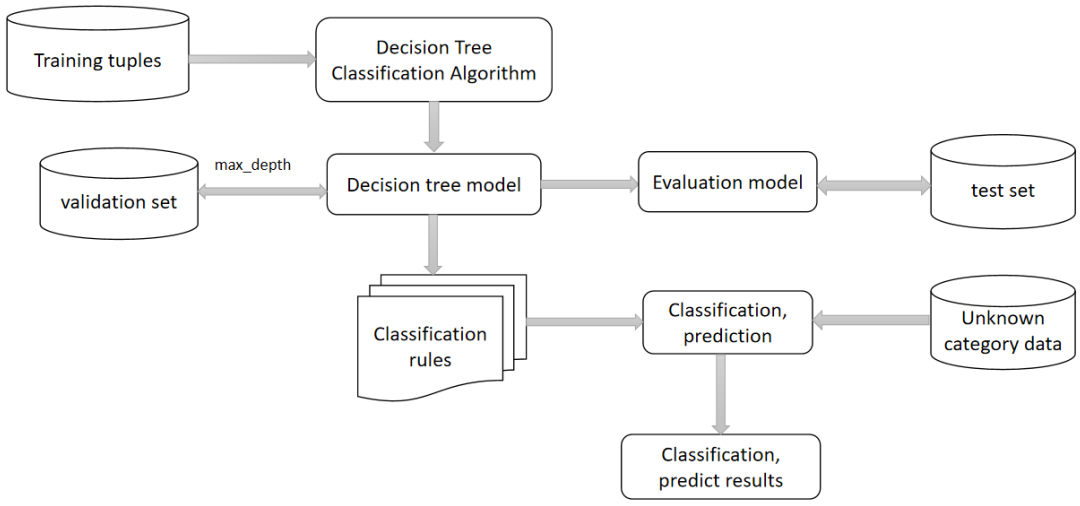

Figure 4.1: Decision tree workflow

The calculation method of impurity reduction caused by the binary division of attribute A is as follows:

$$\Delta Gini(A) = Gini(T) - Gini_A(T)$$

4.1.4 Decision Tree Workflow

The decision tree workflow of this module is shown in Figure 4.1:

In the process of generating the decision number analyzer model, a 5-fold cross-validation GridSearch is used. Here we choose the maximum depth parameter max_depth of the decision tree model, we can traverse the values from 1 to 10, and use the roc-auc score as the evaluation criterion to search for the most suitable max_depth value. We only used single parameter tuning here. GridSearch is a parameter tuning method of exhaustive search. It traverses all candidate parameters, builds the model cyclically, evaluates the effectiveness and accuracy of the model, and selects the best performing parameter as the final result.

In this module, we use the GridSearchCV() function in the Scikit-Learn library to optimize the parameters of the max_depth of the decision tree model (Mujtaba, 2020).

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
```

The above code imports the DecisionTreeClassifier() and GridSearchCV() functions from the Scikit-Learn library.

```
param ={'max_depth': range(1, 10, 1)}
```

The above code specifies the candidate value range of the parameter `max_depth` to be tuned in the decision tree model.

```
model = DecisionTreeClassifier(random_state=123)
```

The above code builds a decision tree model and assigns it to the variable model. The parameter `random_state` sets the random seed number, which means that the random seed number generated by np.random is used to obtain the same result for multiple runs.

```
grid_search = GridSearchCV(estimators,
                    param_grid = param,
                    scoring='roc\_auc', cv=5 )
grid_search.fit(X, y)
```

The above code passes the decision tree model and the candidate value range of the parameters to be tuned into the `GridSearchCV()` function, and sets the scoring parameter to `'roc_auc'`, which means that the `AUC` value of the `ROC` curve is used as the evaluation standard, and the `cv` parameter is set to `5`, which means to proceed `5-fold` cross-validation.

4.1.5 Training the Classifier

The data sets between the three models are the same. The final data set includes 1118 ICU patients. The classification ratio of patients with sepsis to patients without sepsis is 1:1. The split ratio of training set, validation set and test set is 7:1:2.

The decision tree classifier of sepsis is a machine learning model. The parameter max_depth of the classifier is selected using the validation set, the model is trained, and the evaluation standard is ROCAUC corresponding to different max_depth, and then the performance of the selected model is evaluated on the test set.

4.1.6 Validation

The validation set is a sample set used to adjust the parameters of the classifier (Brownlee, 2017). In the decision tree model, the parameter max_depth of the model is selected using the validation set. Different max_depth corresponds to different training models. In order to find the best model, each model is used to predict the validation set and the accuracy of the model is recorded. Select the parameters corresponding to the best model, and this model is the best model.

The classifier uses GridSearchCV to automatically adjust the parameters. Table 4.1 shows the process of adjusting the parameters. Optimizing the maximum number of features of the decision tree parameters max_depth, max_depth parameter selection range: 1-10, step length is 1, five-fold cross-validation selects the best max_depth.

4.1.7 Result

It can be seen from Table 4.1 that the best max_depth is selected as 5, and the accuracy is 66.51%. Different max_depths have little difference in accuracy. In this thesis, we only select the optimal parameters for max_depth. Of course, if necessary, other optimal parameters are also installed with similar steps to search.

Table 4.1: Using GridSearchCV to select max_depth

max_depth	Mean test scores of CV results
1	0.60580864
2	0.62557859
3	0.64481449
4	0.64298507
5	0.65273407
6	0.63272103
7	0.63284729
8	0.62569195
9	0.61699703
10	00.63363817

```
Evaluation metrics on MyDT:
Test Accuracy: 0.6651785714285714
Test Precision: 0.7011494252873564
Test Recall: 0.5545454545454546
Test F1-score: 0.6192893401015228
Test ROC-AUC: 0.663237639553429
```

Figure 4.2: Evaluation metrics of decision tree

```
Best parameters set:
max_depth: 5
```

The Evaluation metrics of decision tree see Figure 4.2. The accuracy is 0.665, the precision is 0.701, the recall is 0.555, the F1-score is 0.619, the ROC-AUC is 0.663. Receiver operating characteristic (ROC) curve for decision tree see Figure 4.3.

4.2 Implementation of LSTM

Many Python packages, such as pybrain, kears, tensorflow, cikit-neuralnetwork, and others (*Python utilized LSTM time series prediction model analysis(Others-Community)*, n.d.), may be used directly to create LSTM models. In this thesis, we use pytorch to construct LSTM.

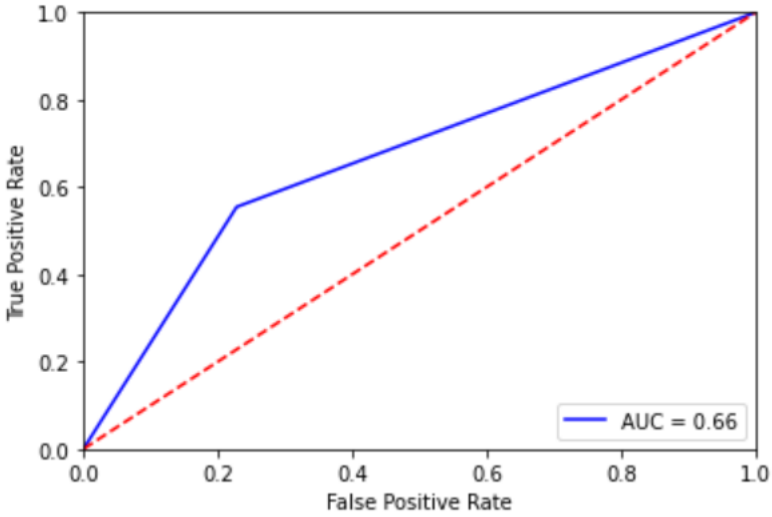

Figure 4.3: Receiver operating characteristic (ROC) curve for decision tree

4.2.1 Model Building

1. Build the network layer, forward propagation forward()

Create an LSTM layer and a linear layer, the LSTM layer extracts features, and the linear layer is used as the final prediction. The number of features corresponding to input_size, our paper is 8, that is, 8 features

```
self.embedding = nn.Linear(in_features=dim_input,
                           out_features=8)
self.rnn = nn.LSTM(input_size=8, hidden_size=4,
                   num_layers=2, batch_first=True)
self.output = nn.Linear(in_features=4, out_features=2)
```

After defining each layer, we finally need to string these together through forward propagation, which involves how to define the forward function. The task of the forward function needs to link the input layer, network layer, and output layer to realize the forward transmission of information. The parameter of forward is the input data, and the return value is the output data.

2. Instantiate the network and create an object of the LSTM() class. Define the

loss function and optimizer, and create an instance of the optimizer.

```
model = MyLSTM(num_features)
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.005)
```

3. Training model, backpropagation backward()

When training the model, you need to put the model in training mode, that is, call model.train().

```
for epoch in range(NUM_EPOCHS):
    train_loss, train_accuracy = train(model,
                                       device,
                                       train_loader,
                                       criterion,
                                       optimizer, epoch)
```

The gradient is cumulative by default, you need to manually initialize or clear the gradient, call:

```
optimizer.zero_grad()
```

Instantiate the model:

```
output = model(input)
```

In the training process, forward propagation generates the output of the network, and calculates the loss value between the output and the actual value. Call loss.backward() to automatically generate the backpropagation gradient, and then use optimizer.step() to execute the optimizer to propagate the gradient back to each network.

```
loss = criterion(output, target)
```

```
#'Model diverged with loss = NaN'
assert not np.isnan(loss.item()),


loss.backward()
optimizer.step()
```

The method of realizing gradient backpropagation is mainly the chain rule of compound functions. Pytorch provides the function of automatic backpropagation. Using the nn toolbox, you don't need to write backpropagation yourself, just let the loss function call backward(). In backpropagation, the optimizer is very important. This type of optimization algorithm updates the parameters by using the gradient values of the parameters.

4. Test and evaluate the model

```
for epoch in range(NUM_EPOCHS):
    valid_loss, valid_accuracy, valid_results =
            evaluate(model, device, valid_loader, criterion)
```

call

```
model.eval()
```

to change the model to test or verification mode, and set the training attribute to false to make the model in the test or verification state.

Then there is the process of forecasting, evaluating, and obtaining relevant evaluation indicators.

4.2.2 LSTM Model Data

The data sets between the three models are the same. The final data set includes 1118 ICU patients. The classification ratio of patients with sepsis to patients without sepsis is

1:1. The split ratio of training set, validation set and test set is 7:1:2. creating a feature sequence data to enter the model.

4.2.3 LSTM Model Basic Parameter Settings

In the experiment, Pytorch is used to implement LSTM. The key steps are mainly data preparation and model construction. The data uses the same data as machine learning. In the model construction, use the nn toolbox, and in the training process, calculate the loss value between the output value and the actual value on the validation set, let's keep the model that has the best accuracy. We will train our model with 20 epochs. Of course, if we want, we can try to use more iterations. We set bath size is 64, At each iteration, it takes 13 readings to train all the data, and the training time, loss and accuracy will be printed. The model uses Adam optimizer. Adam optimizer was proposed by Kingma and other scholars in 2015. Adam optimizer combines the advantages of AdaGrad and RMSProp optimization algorithms(Kingma & Ba, 2015). The First Moment Estimation of the gradient, the mean value of the gradient, and the Second Moment Estimation, the uncentered variance of the gradient are comprehensively considered, and the update step is calculated.

4.2.4 The Output of LSTM Model

Set epochs to 20 and iterative training 20 times. The training set has more than 700 data. Because the batch_size is 64, each iteration needs to be read 13 times. The output results printed during the training process are as follows. The final LSTM model is the model with the highest verification accuracy for 20 iterations.

Table 4.2: Train and Evaluate epochs output(LSTM)

	Epoch	Performance
†[1]	**0:[0/13]**	**Time:**0.016, **Loss:**0.2826, **Accuracy:** 92.188
	0:[10/13]	**Time:**0.012, **Loss:**0.3462, **Accuracy:** 85.227
	Test:	**Time:**0.004, **Loss:**0.3657, **Accuracy:** 84.375
†	**1:[0/13]**	**Time:**0.011, **Loss:**0.3490, **Accuracy:** 84.375
	1:[10/13]	**Time:**0.011, **Loss:**0.3191, **Accuracy:** 82.812
	Test:	**Time:**0.004, **Loss:**0.3667, **Accuracy:** 84.375
†	**2:[0/13]**	**Time:**0.011, **Loss:**0.2960, **Accuracy:** 89.062
	2:[10/13]	**Time:**0.011, **Loss:**0.3114, **Accuracy:** 85.938
	Test:	**Time:**0.008, **Loss:**0.3677, **Accuracy:** 84.375
†	**3:[0/13]**	**Time:**0.017, **Loss:**0.3308, **Accuracy:** 84.375
	3:[10/13]	**Time:**0.013, **Loss:**0.3195, **Accuracy:** 89.062
	Test:	**Time:**0.004, **Loss:**0.3813, **Accuracy:** 84.375
†	**4:[0/13]**	**Time:**0.011, **Loss:**0.3456, **Accuracy:** 85.938
	4:[10/13]	**Time:**0.010, **Loss:**0.3646, **Accuracy:** 84.375
	Test:	**Time:**0.004, **Loss:**0.3718, **Accuracy:** 84.375
†	**5:[0/13]**	**Time:**0.010, **Loss:**0.3633, **Accuracy:** 82.812
	5:[10/13]	**Time:**0.011, **Loss:**0.4036, **Accuracy:** 87.500
	Test:	**Time:**0.006, **Loss:**0.3667, **Accuracy:** 82.812
†	**6:[0/13]**	**Time:**0.011, **Loss:**0.2686, **Accuracy:** 93.750
	6:[10/13]	**Time:**0.011, **Loss:**0.3563, **Accuracy:** 85.938
	Test:	**Time:**0.004, **Loss:**0.3720, **Accuracy:** 84.375
†	**7:[0/13]**	**Time:**0.010, **Loss:**0.4334, **Accuracy:** 81.250
	7:[10/13]	**Time:**0.012, **Loss:**0.5354, **Accuracy:** 75.000
	Test:	**Time:**0.004, **Loss:**0.3687, **Accuracy:** 85.938
†	**8:[0/13]**	**Time:**0.016, **Loss:**0.3699, **Accuracy:** 85.938
	8:[10/13]	**Time:**0.011, **Loss:**0.3028, **Accuracy:** 85.938
	Test:	**Time:**0.004, **Loss:**0.3848, **Accuracy:** 84.375
†	**9:[0/13]**	**Time:**0.015, **Loss:**0.2911, **Accuracy:** 85.938

Continued over page

[1]demographic information of the patient(1-6).

Table 4.2: Tables in... *(continued)*

	Epoch	Performance
	9:[10/13]	**Time:**0.012, **Loss:**0.3823, **Accuracy:** 84.375
	Test:	**Time:**0.004, **Loss:**0.4000, **Accuracy:** 84.375
†	**10:[0/13]**	**Time:**0.010, **Loss:**0.3377, **Accuracy:** 84.375
	10:[10/13]	**Time:**0.011, **Loss:**0.2888, **Accuracy:** 87.500
	Test:	**Time:**0.004, **Loss:**0.3649, **Accuracy:** 85.938
†	**11:[0/13]**	**Time:**0.011, **Loss:**0.3103, **Accuracy:** 90.625
	11:[10/13]	**Time:**0.011, **Loss:**0.3678, **Accuracy:** 82.812
	Test:	**Time:**0.004, **Loss:**0.3630, **Accuracy:** 85.938
†	**12:[0/13]**	**Time:**0.010, **Loss:**0.3849, **Accuracy:** 81.250
	12:[10/13]	**Time:**0.011, **Loss:**0.2469, **Accuracy:** 92.188
	Test:	**Time:**0.004, **Loss:**0.3822, **Accuracy:** 84.375
†	**13:[0/13]**	**Time:**0.010, **Loss:**0.2190, **Accuracy:** 90.625
	13:[10/13]	**Time:**0.012, **Loss:**0.3355, **Accuracy:** 87.500
	Test:	**Time:**0.004, **Loss:**0.3536, **Accuracy:** 84.375
†	**14:[0/13]**	**Time:**0.014, **Loss:**0.3135, **Accuracy:** 87.500
	14:[10/13]	**Time:**0.010, **Loss:**0.2433, **Accuracy:** 92.188
	Test:	**Time:**0.004, **Loss:**0.3812, **Accuracy:** 81.250
†	**15:[0/13]**	**Time:**0.012, **Loss:**0.2866, **Accuracy:** 87.500
	15:[10/13]	**Time:**0.016, **Loss:**0.3394, **Accuracy:** 87.500
	Test:	**Time:**0.004, **Loss:**0.3755, **Accuracy:** 84.375
†	**16:[0/13]**	**Time:**0.010, **Loss:**0.4375, **Accuracy:** 82.812
	16:[10/13]	**Time:**0.013, **Loss:**0.2908, **Accuracy:** 92.188
	Test:	**Time:**0.005, **Loss:**0.3675, **Accuracy:** 85.938
†	**17:[0/13]**	**Time:**0.012, **Loss:**0.2983, **Accuracy:** 87.500
	17:[10/13]	**Time:**0.011, **Loss:**0.3733, **Accuracy:** 82.812
	Test:	**Time:**0.004, **Loss:**0.4158, **Accuracy:** 82.812
†	**18:[0/13]**	**Time:**0.010, **Loss:**0.2587, **Accuracy:** 90.625
	18:[10/13]	**Time:**0.011, **Loss:**0.2884, **Accuracy:** 90.625
	Test:	**Time:**0.004, **Loss:**0.3821, **Accuracy:** 84.375
†	**19:[0/13]**	**Time:**0.011, **Loss:**0.3300, **Accuracy:** 85.938

Continued over page

```
Evaluation metrics on train set:
Test Accuracy: 0.8580562659846548
Test Precision: 0.90625
Test Recall: 0.8035264483627204
Test F1-score: 0.8518024032042723
Test ROC-AUC: 0.9274853609866204
```

Figure 4.4: Evaluation metrics on train set (LSTM)

Table 4.2: Tables in... *(continued)*

Epoch	Performance
19:[10/13]	**Time:**0.011, **Loss:**0.3442, **Accuracy:** 84.375
Test:	**Time:**0.006, **Loss:**0.3777, **Accuracy:** 84.375

4.2.5 The result of LSTM model

The Evaluation metrics of LSTM model on train set see Figure 4.4. The accuracy is 0.858, the precision is 0.906, the recall is 0.804, the F1-score is 0.852, the ROC-AUC is 0.927. Receiver operating characteristic (ROC) curve for decision tree see Figure 4.5.

The Evaluation metrics of LSTM model on validation set see Figure 4.6. The accuracy is 0.8125, the precision is 0.925, the recall is 0.673, the F1-score is 0.779, the ROC-AUC is 0.927. Receiver operating characteristic (ROC) curve for decision tree see Figure 4.7.

The Evaluation metrics of LSTM model on test set see Figure 4.8. The accuracy is 0.866, the precision is 0.908, the recall is 0.809, the F1-score is 0.855, the ROC-AUC is 0.935. Receiver operating characteristic (ROC) curve for decision tree see Figure 4.9.

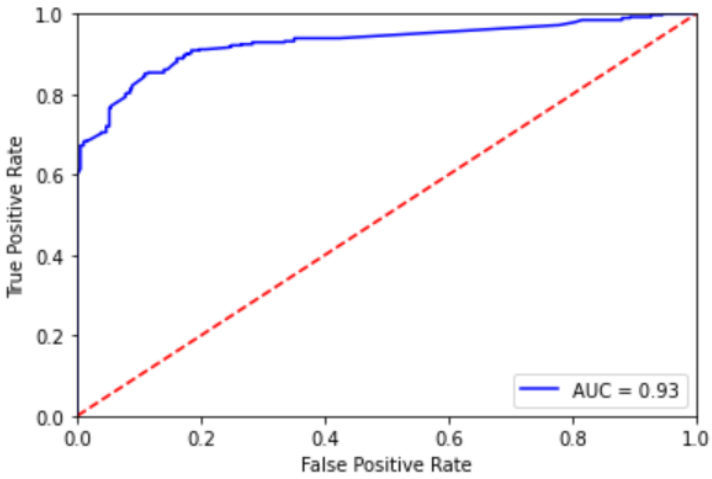

Figure 4.5: ROC curve on train set (LSTM)

```
Evaluation metrics on validation set:
Test Accuracy: 0.8125
Test Precision: 0.925
Test Recall: 0.6727272727272727
Test F1-score: 0.7789473684210527
Test ROC-AUC: 0.8352472089314195
```

Figure 4.6: Evaluation metrics on validation set (LSTM)

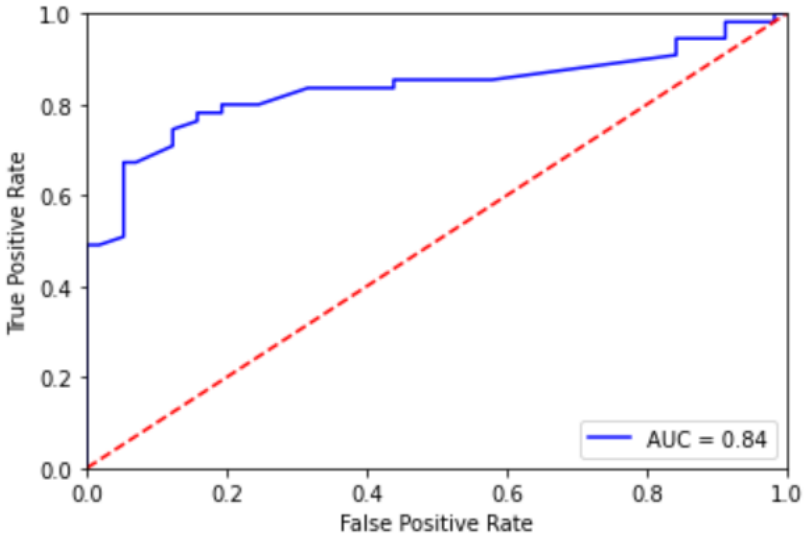

Figure 4.7: ROC curve on validation set (LSTM)

```
Evaluation metrics on test set:
Test Accuracy: 0.8660714285714286
Test Precision: 0.9081632653061225
Test Recall: 0.8090909090909091
Test F1-score: 0.8557692307692307
Test ROC-AUC: 0.9347687400318979
```

Figure 4.8: Evaluation metrics on test set (LSTM)

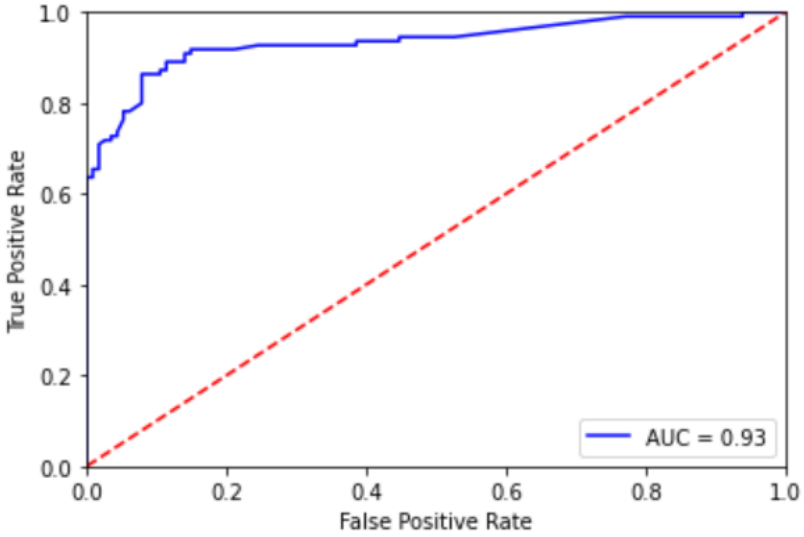

Figure 4.9: ROC curve on test set (LSTM)

4.3 TCN Model to Predict Sepsis

4.3.1 TCN Model for Predict Spesis

The timing convolution network actually transforms the one-dimensional convolution into a model suitable for timing problems (Bai, Kolter & Koltun, 2018). Use a multi-layer network to learn information over a longer time span, and each layer of the network uses a one-dimensional convolution kernel to scan all data at the current time point. The sequence information is passed along the network layer by layer, and finally the prediction result can be obtained. The deeper the network, the longer the information can be learned.

There are two main differences between time series convolution and ordinary one-dimensional convolution:

a: Use dilated convolutions, the higher the layer, the bigger the convolution window and the more holes in it. the dilated convolution can ensure that each hidden layer is the same size as the input sequence, reducing the calculation And increase the receptive field so that the model can learn the information over a longer period of time.

b: Use causal convolution, because time series prediction can only use information before time t to predict the value of time t. Causal convolution restricts the sliding window to ensure that the information after time t will not be used for prediction.

From the above summary, we can see that time series convolution is only a structural innovation of one-dimensional convolution. The time series convolution model is proposed to solve the problem that the RNN model cannot be operated in parallel, and to obtain faster calculation speed. It has comparable performance with RNN in time

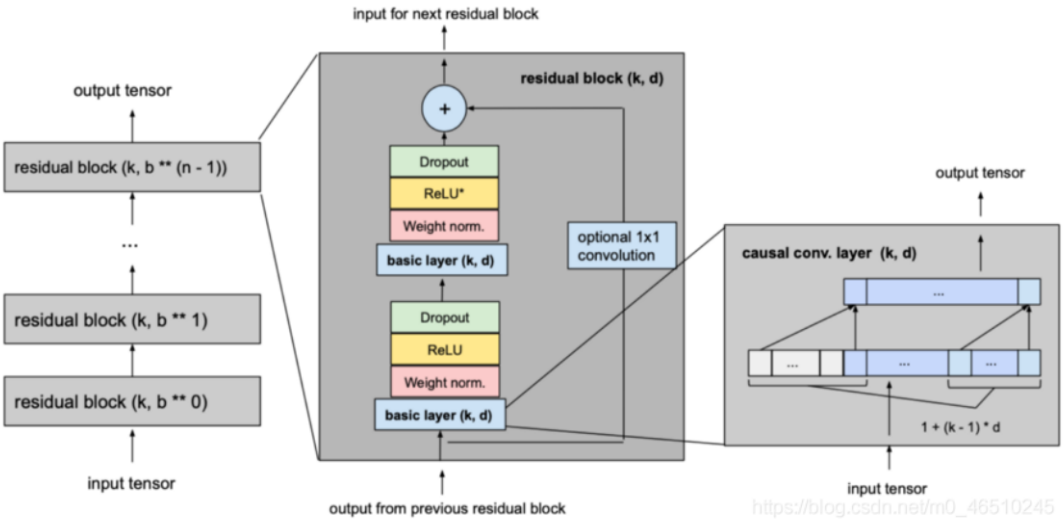

Figure 4.10: Architecture of the TCN model *(Bai, Kolter & Koltun, 2018)*

series prediction (Kechyn, Yu, Zang & Kechyn, 2018). This thesis uses TCN and LSTM for comparison.

Figure 4.10 shows our TCN model, where l is equal to input_length, k is equal to kernel_size, b is equal to dilation_base, $k \geq b$, and n is the minimum number of residual blocks for complete historical coverage (Bai, Kolter & Koltun, 2018).

$$n = \lceil log_b(\frac{(l-1)\cdot(b-1)}{(k-1)\cdot 2} + 1)\rceil$$

4.3.2 TCN Model Data

The data sets between the three models are the same. The final data set includes 1118 ICU patients. The classification ratio of patients with sepsis to patients without sepsis is 1:1. The split ratio of training set, validation set and test set is 7:1:2. creating feature sequence data to enter, train, validate and test the model.

However, it should be noted that the TCN data cannot be a variable-length sequence, and the same sequence must be entered. Therefore, when processing the data, the

maximum length is 13, which is 12 hours of historical data and a current data.

```
batch_seq, batch_label = zip(*batch)


num_features = batch_seq[0].shape[1]
seq_lengths = list(map(lambda patient_tensor:
patient_tensor.shape[0], batch_seq))
max_length = 13
```

4.3.3 Pytorch Implements TCN

The TCN of this thesis is implemented based on pytorch, import related packages.

```
import torch
import torch.nn as nn
```

Implement causal convolution

The class that implements causal convolution inherits from the class nn.Module. Then use forward propagation to turn the tensor into a form of continuous distribution in memory, and tensor.contiguous() will return the same tensor with contiguous memory.

```
class Chomp1d(nn.Module):
    def _init_(self, chomp_size):
        ...
    def forward(self, x):
        return x[:, :, :-self.chomp_size].contiguous()
```

Implement the residual module

The class that implements residual module inherits from the class nn.Module.

```
class TemporalBlock(nn.Module):
    def _init_(self, n_inputs, n_outputs, kernel_size,
                    stride, dilation, padding, dropout=0.2):
```

Parameter description:

n_inputs: the number of input channels

n_outputs: the number of output channels

kernel_size: convolution kernel size

stride: stride, generally 1

dilation: expansion coefficient

padding: padding factor

dropout: dropout ratio

Define the convolutional layer, and implement causal convolution according to the output of the convolutional layer and the padding size, crop the extra padding part, and maintain the output time step. Then add the activation function and dropout to the previous output to complete a convolution. In this way, multiple convolutional layers can be defined.

```
self.conv = weight_norm(nn.Conv1d(n_inputs, n_outputs,
                        kernel_size, stride=stride,
                        padding=padding, dilation=dilation))
self.chomp = Chomp1d(padding)
self.relu = nn.ReLU()
self.dropout= nn.Dropout(dropout)
```

Time convolutional network architecture

The class that implements TCN also inherits from the class nn.Module.

```
class TemporalConvNet(nn.Module):
    def_init_(self, num_inputs, num_channels,
            kernel_size=2, dropout=0.2):
```

Parameter description:

num_inputs: the number of input channels

num_channels: the number of hidden_channels in each layer.

kernel_size: convolution kernel size

dropout: drop_out ratio

4.3.4 TCN Model Basic Parameter Settings

In the experiment, Pytorch is used to implement TCN. The key steps are mainly data preparation and model construction. The data uses the same data as machine learning and LSTM. In the model construction, use the nn toolbox.

When training the model, we only specify the first part of the training series as target_series, because we don't want to predict the assistant time series we added earlier. We tried several different hyper-parameter combinations, but most of the values were chosen arbitrarily. finally

```
epochs=20,


input_size=13,
output_size=2,


num_channels = [32, 64, 128]


dropout=0.2,
kernel_size=2,
```

During the training process, the loss value between the output value and the actual value on the validation set is calculated, allowing us to maintain the model with the best accuracy. We will train our model with 20 epochs. We set the bath_size to 64, and each iteration requires 13 readings to train all the data, and the training time, loss, and accuracy will be printed out. The model uses Adam optimizer.

```
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=0.005)
```

4.3.5 The Output of TCN Model

Same to LATM , TCN also iterative training 20 times. The output results printed during the training process are as follows. The final TCN model is the model with the highest verification accuracy for 20 iterations.

Table 4.3: Train and Evaluate epochs output(TCN)

	Epoch	Performance
†[2]	**0:[0/13]**	**Time:**0.030, **Loss:**0.3899, **Accuracy:** 84.375
	0:[10/13]	**Time:**0.027, **Loss:**0.4046, **Accuracy:** 79.688
	Test:	**Time:**0.007, **Loss:**0.3438, **Accuracy:** 84.375
†	**1:[0/13]**	**Time:**0.026, **Loss:**0.3684, **Accuracy:** 82.812
	1:[10/13]	**Time:**0.020, **Loss:**0.3979, **Accuracy:** 85.511
	Test:	**Time:**0.007, **Loss:**0.3454, **Accuracy:** 85.938
†	**2:[0/13]**	**Time:**0.020, **Loss:**0.3912, **Accuracy:** 89.062
	2:[10/13]	**Time:**0.021, **Loss:**0.2704, **Accuracy:** 87.500
	Test:	**Time:**0.008, **Loss:**0.3659, **Accuracy:** 84.375
†	**3:[0/13]**	**Time:**0.020, **Loss:**0.3004, **Accuracy:** 87.500

Continued over page

[2]demographic information of the patient(1-6).

Table 4.3: Tables in... *(continued)*

	Epoch	Performance
	3:[10/13]	**Time:**0.022, **Loss:**0.3355, **Accuracy:** 75.000
	Test:	**Time:**0.007, **Loss:**0.3374, **Accuracy:** 85.938
†	**4:[0/13]**	**Time:**0.024, **Loss:**0.3045, **Accuracy:** 89.062
	4:[10/13]	**Time:**0.020, **Loss:**0.3471, **Accuracy:** 85.938
	Test:	**Time:**0.007, **Loss:**0.3891, **Accuracy:** 82.812
†	**5:[0/13]**	**Time:**0.027, **Loss:**0.2480, **Accuracy:** 90.625
	5:[10/13]	**Time:**0.020, **Loss:**0.2812, **Accuracy:** 89.062
	Test:	**Time:**0.007, **Loss:**0.3864, **Accuracy:** 82.812
†	**6:[0/13]**	**Time:**0.022, **Loss:**0.2821, **Accuracy:** 85.938
	6:[10/13]	**Time:**0.021, **Loss:**0.3265, **Accuracy:** 87.500
	Test:	**Time:**0.008, **Loss:**0.4050, **Accuracy:** 84.375
†	**7:[0/13]**	**Time:**0.021, **Loss:**0.3992, **Accuracy:** 82.812
	7:[10/13]	**Time:**0.023, **Loss:**0.3124, **Accuracy:** 89.062
	Test:	**Time:**0.007, **Loss:**0.3318, **Accuracy:** 84.375
†	**8:[0/13]**	**Time:**0.023, **Loss:**0.3486, **Accuracy:** 84.375
	8:[10/13]	**Time:**0.020, **Loss:**0.3593, **Accuracy:** 84.375
	Test:	**Time:**0.007, **Loss:**0.3582, **Accuracy:** 81.250
†	**9:[0/13]**	**Time:**0.020, **Loss:**0.3395, **Accuracy:** 82.812
	9:[10/13]	**Time:**0.020, **Loss:**0.3119, **Accuracy:** 85.938
	Test:	**Time:**0.007, **Loss:**0.3682, **Accuracy:** 81.250
†	**10:[0/13]**	**Time:**0.020, **Loss:**0.2992, **Accuracy:** 87.500
	10:[10/13]	**Time:**0.021, **Loss:**0.2829, **Accuracy:** 89.062
	Test:	**Time:**0.008, **Loss:**0.3545, **Accuracy:** 84.375
†	**11:[0/13]**	**Time:**0.022, **Loss:**0.2995, **Accuracy:** 87.500
	11:[10/13]	**Time:**0.020, **Loss:**0.2020, **Accuracy:** 92.188
	Test:	**Time:**0.007, **Loss:**0.3473, **Accuracy:** 82.812
†	**12:[0/13]**	**Time:**0.020, **Loss:**0.2584, **Accuracy:** 89.062
	12:[10/13]	**Time:**0.025, **Loss:**0.3184, **Accuracy:** 85.938
	Test:	**Time:**0.007, **Loss:**0.3115, **Accuracy:** 81.250
†	**13:[0/13]**	**Time:**0.024, **Loss:**0.2679, **Accuracy:** 90.625

Continued over page

Table 4.3: Tables in... *(continued)*

	Epoch	Performance
	13:[10/13]	**Time:**0.022, **Loss:**0.3830, **Accuracy:** 85.938
	Test:	**Time:**0.008, **Loss:**0.3536, **Accuracy:** 81.250
†	**14:[0/13]**	**Time:**0.020, **Loss:**0.2881, **Accuracy:** 92.188
	14:[10/13]	**Time:**0.025, **Loss:**0.2785, **Accuracy:** 89.062
	Test:	**Time:**0.007, **Loss:**0.3694, **Accuracy:** 82.812
†	**15:[0/13]**	**Time:**0.021, **Loss:**0.3129, **Accuracy:** 84.375
	15:[10/13]	**Time:**0.021, **Loss:**0.3012, **Accuracy:** 84.375
	Test:	**Time:**0.007, **Loss:**0.3578, **Accuracy:** 82.812
†	**16:[0/13]**	**Time:**0.020, **Loss:**0.2579, **Accuracy:** 85.938
	16:[10/13]	**Time:**0.020, **Loss:**0.2990, **Accuracy:** 87.500
	Test:	**Time:**0.007, **Loss:**0.3992, **Accuracy:** 79.688
†	**17:[0/13]**	**Time:**0.021, **Loss:**0.2962, **Accuracy:** 87.500
	17:[10/13]	**Time:**0.019, **Loss:**0.3350, **Accuracy:** 87.500
	Test:	**Time:**0.008, **Loss:**0.3778, **Accuracy:** 84.375
†	**18:[0/13]**	**Time:**0.020, **Loss:**0.2806, **Accuracy:** 89.062
	18:[10/13]	**Time:**0.021, **Loss:**0.3633, **Accuracy:** 84.375
	Test:	**Time:**0.008, **Loss:**0.3527, **Accuracy:** 84.375
†	**19:[0/13]**	**Time:**0.021, **Loss:**0.4143, **Accuracy:** 76.562
	19:[10/13]	**Time:**0.021, **Loss:**0.2290, **Accuracy:** 92.188
	Test:	**Time:**0.007, **Loss:**0.3529, **Accuracy:** 84.375

4.3.6 The result of TCN model

The Evaluation metrics of TCN model on train set see Figure 4.11. The accuracy is 0.850, the precision is 0.932, the recall is 0.841, the F1-score is 0.884, the ROC-AUC is 0.938. Receiver operating characteristic (ROC) curve for decision tree see Figure 4.12.

The Evaluation metrics of LSTM model on validation set see Figure 4.13. The accuracy is 0.813, the precision is 0.905, the recall is 0.691, the F1-score is 0.784, the ROC-AUC is 0.841. Receiver operating characteristic (ROC) curve for decision tree

```
Evaluation metrics on train set:
Test Accuracy: 0.8849104859335039
Test Precision: 0.9322493224932249
Test Recall: 0.8410757946210269
Test F1-score: 0.884318766066838
Test ROC-AUC: 0.93753154558624
```

Figure 4.11: Evaluation metrics on train set (TCN)

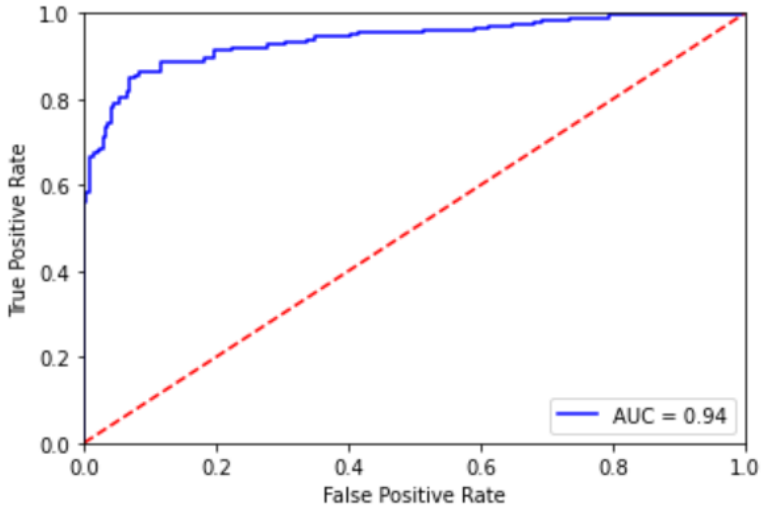

Figure 4.12: ROC curve on train set (TCN)

see Figure 4.14.

The Evaluation metrics of TCN model on test set see Figure 4.15. The accuracy is 0.893, the precision is 0.913, the recall is 0.864, the F1-score is 0.888, the ROC-AUC is 0.944. Receiver operating characteristic (ROC) curve for decision tree see Figure 4.16.

```
Evaluation metrics on validation set:
Test Accuracy: 0.8125
Test Precision: 0.9047619047619048
Test Recall: 0.6909090909090909
Test F1-score: 0.7835051546391754
Test ROC-AUC: 0.8405103668261563
```

Figure 4.13: Evaluation metrics on validation set (TCN)

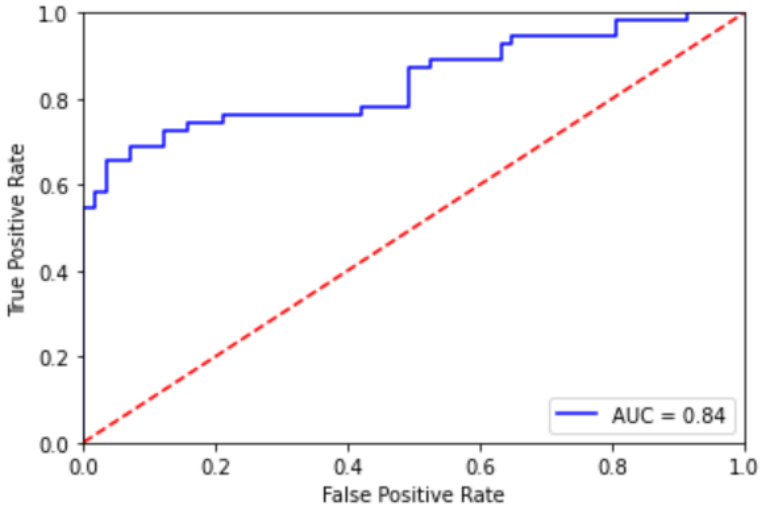

Figure 4.14: ROC curve on validation set (TCN)

```
Evaluation metrics on test set:
Test Accuracy: 0.8928571428571429
Test Precision: 0.9134615384615384
Test Recall: 0.8636363636363636
Test F1-score: 0.8878504672897196
Test ROC-AUC: 0.9442583732057417
```

Figure 4.15: Evaluation metrics on test set (TCN)

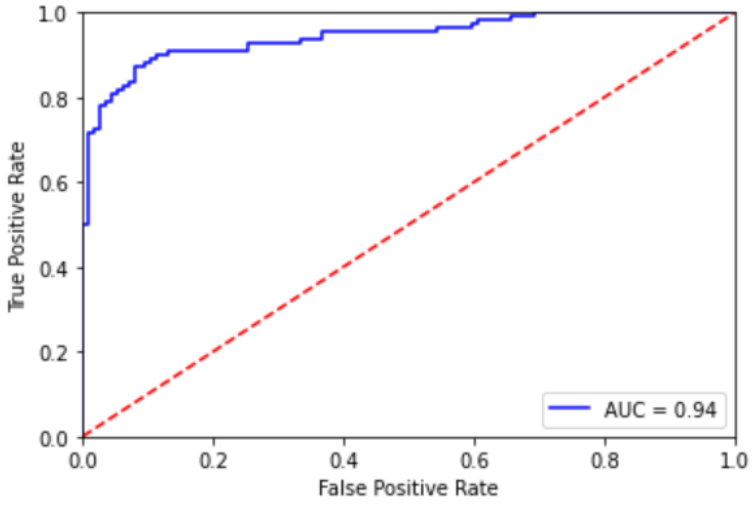

Figure 4.16: ROC curve on test set (TCN)

Chapter 5

Discussion

This chapter first compares the three models of decision tree, LSTM, and TCN to predict the performance of sepsis on the test data set. Then discusses the results in depth and analyzes the advantages and limitations of the TCN model.

5.1 Model Performance Comparison

We compared the performance of the TCN model with the performance of the decision tree classifier algorithm (machine learning) and the LSTM algorithm (deep learning). The three methods apply the same labeling data. Table 5.1 shows all the performance index values of the decision tree, LSTM and TCN models. However, in Chapter 3, in the selection of model evaluation indicators, we choose ROCAUC as the decisive factor for evaluating the pros and cons of the model. So the ROC curve in Figure 5.1 of the three models is drawn.

We mainly use the area under the AUC-ROC characteristic curve to express the performance of the model. The specific meaning of AUC-ROC can be seen in Appendix A. Performance comparison can draw the following conclusions:

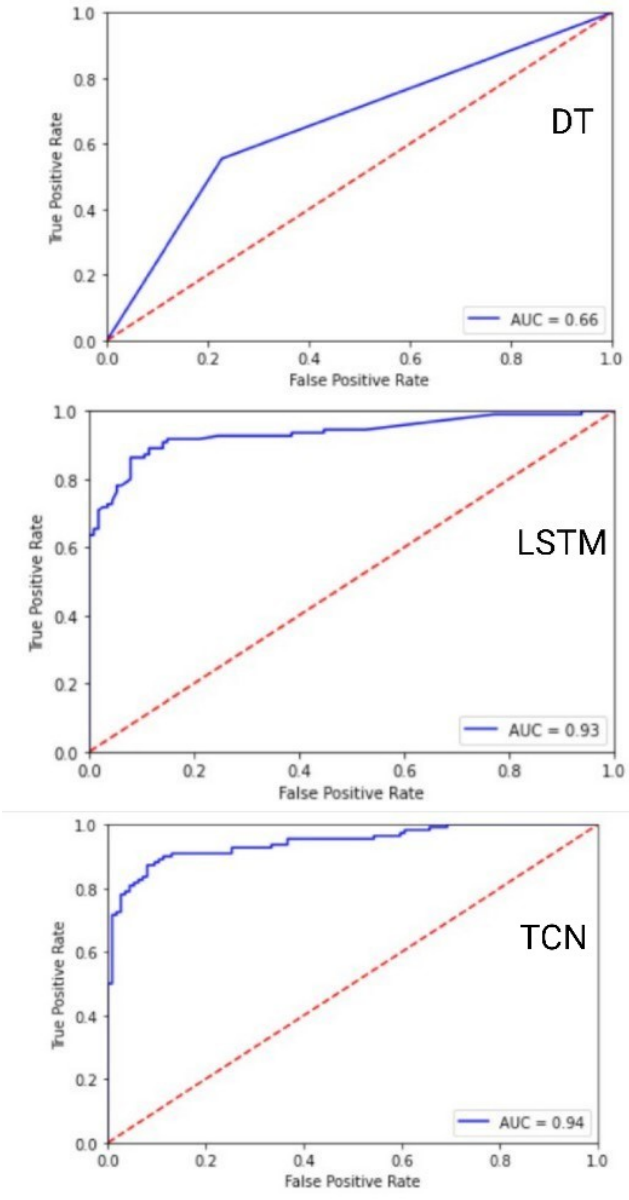

Figure 5.1: model ROC curve comparison

Table 5.1: Comparison of results for all models

	Decison Tree	**LSTM**	**TCN**
Accuracy	0.6652	0.8661	0.8929
Precision	0.7011	0.9082	0.9135
Recall	0.5545	0.8091	0.8636
F1-score	0.6193	0.8558	0.8879
AUC-ROC	0.6632	0.9348	0.9443

- Among 3 classifiers, TCN performs best while Descision Tree performs worst.
- Machine learning methods can help predict the onset of sepsis.
- The TCN and LSTM model believes that the time dependence of sepsis will lead to significantly higher predictors.
- The TCN model is more optimized than LSTM.

The decision tree algorithm is based on a heuristic algorithm, with a small amount of calculation, and the constructed tree structure is also easy to understand, which can show which features are more important. But decision trees tend to ignore the correlation between data. Our data is time series data, the value is measured every 1 hour, and there is a time sequence between the data. Therefore, the decision tree classifier performed relatively poorly on our data set.

LSTM is an excellent variant of RNN, it is very suitable for dealing with timing problems, so the LSTM model also shows high performance. TCN is a new type of algorithm that can be used to solve time series prediction. TCN solves the concurrency problem of LSTM. In our model, TCN beats the LSTM model by a relatively small advantage. The accuracy of deep learning depends on the amount of data, and the amount of data here is not particularly large. Compared with LSTM, the speed of TCN has been greatly improved, and the performance is better than LSTM, which is more suitable for production environment use.

5.2 Discussion

In this study, we used 8 vital sign variables, 559 ICU with sepsis and 559 ICU without sepsis. After data splitting, the data of 780 patients were used as training data. The data set is not particularly small data set. It can be seen from Table 5.1 that the TCN model we proposed achieves an accuracy of 94.4%, which is better than the machine learning and LSTM models implemented by this thesis, but further performance optimization is required. Moor et al. first proposed the use of the TCN model to predict sepsis (Moor et al., 2019). Although we have also established a TCN-based predictive sepsis model, our model is different from the model proposed by Moor. Our classifier is not based on a multi-task Gaussian process, nor does it use a dynamic time warped k-nearest neighbor classifier. We all use data from MIMIC-III, but the labels and patient data used are different. We both do early warning analysis to compare different prediction ranges in the hours leading up to sepsis onset. Therefore, in the next analysis direction, we mainly optimize the TCN, focusing on the comparison of the MPG-TCN model proposed by Moor, in order to achieve better classification performance than previous researchers.

It is worth noting that the data set we use is a balanced data set, and the AUC-ROC indicator can be used directly to measure the model. However, there are many previous studies. The data set used by their model is an unbalanced data set. The ROC graph cannot reflect the real classification performance of the classifier of the unbalanced data set, because this will lead to a misleading interpretation of the sensitivity of the model (Futoma et al., 2017). Therefore, in the comparison with the first, we cannot simply compare the accuracy and ROC-AUC, we need to consider the overall data set. For unbalanced data sets, we use AUC-PRC. The PRC chart gives a more precise forecast of the model's future classification performance because it calculates the proportion of true positives in the positive prediction (Futoma et al., 2017).

For the other models mentioned in the literature review, our TCN also has an

absolute advantage if we look at the ROC-AUC value alone, but this comparison is of little significance. Because different documents use different data sets, and the experimental scenarios are different. It is for this reason that this thesis not only developed the TCN model, but also the decision tree and LSTM two benchmark models. The three models are trained on the same data set, the best model is selected with the same verification set, and the performance is tested on the same test set to compare performance.

Our research has some advantages and limitations, which are discussed as follows:

5.2.1 Benefits

- Three models with different methods are established. The training data and test data used by the models are the same, so the performance comparison of the three models can reflect the advantages of the model itself.
- Data set localization, including patient's vital signs data set and SOFA labeled data.
- Patient warning can extend the window of meaningful clinical intervention.

5.2.2 Limitations

- The current data features only include vital signs (vitals), without considering the result data (labs) of laboratory tests, which is a very important part of data in medicine.
- The current model is basic, and only a simple hyper-parameter selection is made. Further new technologies will be introduced to improve the performance of the model, such as attention.

Chapter 6

Conclusion

This chapter summarizes the entire thesis research, and looks forward to the future, and proposes future research directions in this field.

6.1 Conclusion

Sepsis is a disease in which a person's immune system produces a significant amount of inflammatory chemicals in order to combat infections caused by microorganisms. Fluid can build in the tissues as a result of a significant number of chemicals being released, leading to organ dysfunction. In Sepsis-3, infected patients with SOFA score higher than 2 are identified as sepsis patients. Sepsis is one of the main causes of mortality in hospitals around the world. Studies have shown that the earlier treatment can significantly improve the survival rate, therefore, prediction is more important than detection. If ICU patients can be accurately predicted to develop sepsis in the next few hours, it can help ICU doctors make the best clinical decisions, thereby improving the clinical outcomes of patients with high-risk sepsis and increasing the survival rate of patients. Early and accurate prediction can prevent the lasting effects of sepsis on the patient's body. The goal of our thesis is to use 12-hours observational health data to

predict whether the sepsis will occur in following 6 hours, which is a classification problem.

So far, people have done in-depth research on the problem of sepsis prediction, and many models for predicting sepsis have been developed. These models include machine learning models and some basic deep learning models. These models are based on MIMIC database, and some are based on other data sources. With the development of deep learning models, deep learning models have increasingly replaced machine learning models. In the research of this thesis, we used a deep time convolutional network to predict sepsis, and established a benchmark model, a decision tree machine learning model and a basic LSTM deep learning model. This thesis has completed several tasks:

- This thesis first investigated the background of sepsis prediction and found that early and accurate prediction of sepsis is very important to reality. Then we also investigated the development of the definition of sepsis. In this thesis, we use the spesis-3 standard proposed in 2016 to quantify whether a patient has sepsis. In addition, we also investigated the current development of sepsis prediction models, which mainly include machine learning models and deep learning models. Most of the models are based on the MIMIC database, and more of them are based on the MIMIC-III database. So what we have achieved in this thesis is to build a deep learning TCN model based on the MIMIC-III database using 12 hours of observation to predict whether sepsis will occur in following 6 hours.
- By consulting the literature, we learned about the development of researchers in the field of sepsis prediction. The focus is on the application of deep learning models in predicting sepsis. Finally, the development of MIMIC-III database and its table structure are investigated. This part of the content provides the direction for our research in this thesis. We will establish a TCN deep learning model to

predict sepsis, and establish two benchmark models for model evaluation.

- Learned the theoretical knowledge related to ML(decision trees), LSTM and TCN, and learned the related theoretical knowledge of model evaluation. This part of the content provides a strong theoretical foundation for the establishment of the three models.
- It has been achieved to obtain sepsis-related data from MIMIC-III, and construct a characteristic sequence model for the model. First, import the MIMIC-III database file into the postgreSQL database, and use pgAdmin4 to manage it, use MIMIC-III code base and Spesis-mimic code base to generate sepsis-related data tables, including pivoted_vital, pivoted_score, spesis3 and other database tables, export csv for data processing raw data. Then generate SOFA timeline information and retrieve sepsis in ICU and the corresponding onset time, extract relevant features and mark the data formed as a cleaned data set, and then process the data into serialized data for training, verifying and testing the model.
- In feature extraction, we extracted 8 vital signs (heart rate,body temperature,Systolic blood pressure,Diastolic blood pressure,Mean blood pressure,Respiratory rate,Pulse oximetry and glucose) and 2 basic variables (age, gender), and marked sepsis for each record.
- Implements a machine learning benchmark model, decision tree. Experiments have shown that we can use machine learning models to predict sepsis, and the data set is also serialized data. Different max_depths have little difference in accuracy. The best max_depth is selected as 5, and the accuracy of test dataset is 66.52%, the ROC-AUC is 66.32%.
- Implements a deep learning benchmark model, LSTM. The model uses Adam optimizer. Set batch_size to 64. We will train our model with 20 epochs. Use sequence data to train, verify and test the model, and finally achieve a relatively high level of training data set, validation data set and test data set Accuracy. The

accuracy and ROC-AUC of test sets are 88.61% and 93.48%.

- Temporal convolutional networks can handle serialized data well, and only use previous data to generate future data. Therefore, TCN is very suitable to predict whether sepsis will occur in the next 6 hours. First, the cleaned data set obtained by data processing was used to construct feature serialized data, and then in this study, we used a deep time convolutional network to predict sepsis. The acquired feature sequence data is input into the model. For patients, relatively high accuracy and ROC-AUC of 89.3% and 94.4% were achieved respectively.
- Evaluate the Accuracy, Precision, Recall, F1-score and AUC-ROC of each model. Experiments demonstrate that the model we offer is a useful tool that can be used to predict sepsis in an automated diagnostic tool.

6.2 Future Work

In this thesis, a machine learning model (decision tree classifier) and two deep learning models (LSTM and TCN) are designed to predict the possibility of patients in the intensive care unit suffering from sepsis in the next 6 hours. The MIMIC code repository and the Spesis-mimic code repository provide many benefits in terms of source code distribution and enhanced reproducibility. As a result, we obtained data related to sepsis prediction from the postgreSQL database where MIMIC-III data was deployed, labeled, split, constructed sequence data, and trained different models. And through comparison with each other, it is proved that the TCN-based deep learning model algorithm can effectively improve the performance. Although the research has achieved initial results, there is still much room for improvement.

For our future work, we need to improve both the data set and the model.

1. For our current data, we want to explore how to improve the predictive performance

of predicting sepsis when the amount of data is small. This part explores how to improve the prediction performance by improving the model algorithm. The TCN model in this paper is the most basic realization of TCN theory. Next, I will explore the introduction of Multitask Gaussian Process (MGP) to the TCN model. MGP is a non-parametric Bayesian model. Given an hourly interval time series and an irregularly sampled medical time series, the MGP layer may produce a set of posterior predictions for each feature, which can subsequently be fed into the classification model (Moor et al., 2019). MPG uses an approximate Bayesian algorithm to estimate a set of accurate posterior distributions through prior distributions. The posterior distribution data is used as the data set of the training model, which can improve the performance of the classification model. In addition, we also plan to introduce the attention mechanism into the TCN model (Liang, Ke, Zhang, Yi & Zheng, 2018; Guo, Lin, Feng, Song & Wan, 2019), which can effectively identify dynamic dependencies, and provide early warning of sepsis with more precision and high interpretation.

2. In addition, we intend to increase the model's performance by training it with a bigger data set. In order to do this, we need to explore other solutions for missing items in the MIMIC-III data. As well as explore other solutions for the problem of highly imbalanced data sets. For data imbalance, I plan to explore the use of random resampling to solve the problem of data set imbalance.

3. Then we will consider more features, such as drug records, laboratory records, etc. Our current labeling data includes 8 vital signs and 2 basic demographic characteristics. MIMIC-III includes patient demographic information, diagnostic information, laboratory test information, medical imaging information, vital signs, etc., Two major feature clusters stand out: vital signs (vitals) and laboratory results (labs). MIMIC-III has a wealth of data resources, and we only pay attention to vital signs, not other data such as laboratory results. So we can use more features

to strike a better balance between accuracy and recall. Better feature selection can help us better grasp the properties of the data and enhance the model's performance.

4. Use different combinations of observation windows and prediction windows. Our goal in this thesis is to use 12-hours observational health data to predict whether sepsis will occur in following 6 hours. The observation window is 12 hours, and the forecast window is 6 hours. In future explorations, we will test different combinations, weigh performance and window size, and choose the best window combination.

5. In addition, we plan to computerize the TCN-based sepsis prediction tool to predict the future risk of sepsis in ICU patients and integrate it into the current ICU information system, enabling early warning and clinical decision support. In future studies, I will prospectively evaluate the efficacy of predictive models and systems and examine whether it can improve clinical practice. MIMIC-III is linked to the social security database, allowing it to track patient follow-up time and outcomes, which is extremely important for us to carry out long-term prognosis research.

References

Acharya, U. R., Fujita, H., Oh, S. L., Raghavendra, U., Tan, J. H., Adam, M., ... Hagiwara, Y. (2018). Automated identification of shockable and non-shockable life-threatening ventricular arrhythmias using convolutional neural network. *Future Generation Computer Systems*, *79*, 952–959. doi: 10.1016/j.future.2017.08.039

Angus, D. C., Linde-Zwirble, W. T., Lidicker, J., Clermont, G., Carcillo, J. & R., P. M. (2001). Epidemiology of severe sepsis in the united states: analysis of incidence, outcome, and associated costs of care. *Crit Care Med*, *29*, 1303–1310. doi: 10.1097/00003246-200107000-00002

Bahdanau, D., Cho, K. & Bengio, Y. (2016, May). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*. Retrieved 2021-07-08, from `http://arxiv.org/abs/1409.0473` (arXiv: 1409.0473)

Bai, S., Kolter, J. Z. & Koltun, V. (2018, April). An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv:1803.01271 [cs]*. Retrieved 2021-08-12, from `http://arxiv.org/abs/1803.01271` (arXiv: 1803.01271)

Bai, S., Zico Kolter, J. & Koltun.v. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271 [cs.LG]*.

Bone, R. C., Balk, R. A., Cerra, F. B., Dellinger, R. P., Fein, A. M., Knaus, W. A., ... Sibbald, W. J. (1992). Definitions for sepsis and organ failure and guidelines for the use of innovative therapies in sepsis. *Chest*, *101*, 1644–1655. doi: https://doi.org/10.1378/chest.101.6.1644

BorgwardtLab/mgp-tcn. (2021, July). Machine Learning and Computational Biology Lab. Retrieved 2021-07-08, from `https://github.com/BorgwardtLab/mgp-tcn` (original-date: 2019-04-16T08:12:27Z)

Brownlee, J. (2017, July). *What is the Difference Between Test and Validation Datasets?* Retrieved 2021-08-23, from `https://machinelearningmastery.com/difference-test-validation-datasets/`

Calvert, J. S., Price, D. A., Chettipally, U. K., Barton, C. W., Feldman, M. D., Bruining, H., ... Thijs, L. G. (2016). A computational approach to early sepsis detection. *Intensive Care Medicine*, *22*, 707–710. doi: https://doi.org/10.1007/bf01709751

Carrara, M., Baselli, G. & Ferrario, M. (2015, August). Mortality prediction in septic shock patients: Towards new personalized models in critical care. In *2015*

37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 2792–2795). (ISSN: 1558-4615) doi: 10.1109/EMBC.2015.7318971

Celeste, M., Torio, P. & Brian J. Moore, p. (2013). *National inpatient hospital costs: The most expensive conditions by payer, 2013.* Agency for Healthcare Research and Quality. Retrieved May 2016, from https://www.hcup-us.ahrq.gov/reports/statbriefs/sb204-Most-Expensive-Hospital-Conditions.pdf

Celeste, M., Torio, P. & Brian J. Moore, p. (2017). *National inpatient hospital costs: The most expensive conditions by payer, 2017.* Agency for Healthcare Research and Quality. Retrieved July 2020, from https://www.hcup-us.ahrq.gov/reports/statbriefs/sb261-Most-Expensive-Hospital-Conditions-2017.pdf

Cohen, L., Manion, L. & Morrison, K. (2000). *Research Methods in Education* (5th ed.). London: Routledge. doi: 10.4324/9780203224342

Delahanty, R. J., Alvarez, J., Flynn, L. M., Sherwin, R. L. & Jones, S. S. (2019, April). Development and Evaluation of a Machine Learning Model for the Early Identification of Patients at Risk for Sepsis. *Annals of Emergency Medicine*, *73*(4), 334–344. Retrieved 2021-07-07, from https://www.sciencedirect.com/science/article/pii/S0196064418314860 doi: 10.1016/j.annemergmed.2018.11.036

Desautels, T., Calvert, J., Hoffman, J. & Jay, M. e. a. (2016). Prediction of sepsis in the intensive care unit with minimal electronic health record data: A machine learning approach. *JMIR Medical Informatics*, *4*. doi: https://doi.org/10.1056/nejmoa022139

Dremsizov, T., Kellum, J. & Angus, D. (2004, May). Incidence and Definition of Sepsis and Associated Organ Dysfunction. *The International Journal of Artificial Organs*, *27*(5), 352–359. Retrieved 2021-07-17, from https://doi.org/10.1177/039139880402700503 (Publisher: SAGE Publications Ltd STM) doi: 10.1177/039139880402700503

Dürichen, R., Pimentel, M. A. F., Clifton, L., Schweikard, A. & Clifton, D. A. (2015, January). Multitask Gaussian Processes for Multivariate Physiological Time-Series Analysis. *IEEE Transactions on Biomedical Engineering*, *62*(1), 314–322. (Conference Name: IEEE Transactions on Biomedical Engineering) doi: 10.1109/TBME.2014.2351376

Elsaraiti, M. & Merabet, A. (2021, January). Application of Long-Short-Term-Memory Recurrent Neural Networks to Forecast Wind Speed. *Applied Sciences*, *11*(5), 2387. Retrieved 2021-08-12, from https://www.mdpi.com/2076-3417/11/5/2387 (Number: 5 Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/app11052387

Fleischmann, C., Scherag, A., Adhikari, N. K. J., Hartog, C. S., Tsaganos, T., Schlattmann, P., ... Reinhart, K. (2016). The epidemiology of sepsis in the united states from 1979 through 2000. *American Journal of Respiratory and Critical Care Medicine*, *193*, 259–272. doi: https://doi.org/10.1164/rccm.201504-0781oc

Futoma, J., Hariharan, S., Heller, K., Sendak, M., Brajer, N., Clement, M., ... O'Brien, C. (2017, November). An Improved Multi-Output Gaussian Process RNN with Real-Time Validation for Early Sepsis Detection. In *Machine Learning for Healthcare Conference* (pp. 243–254). PMLR. Retrieved 2021-07-07, from http://proceedings.mlr.press/v68/futoma17a.html (ISSN: 2640-3498)

Ghosh, S., Li, J., Cao, L. & Ramamohanarao, K. (2017, February). Septic shock prediction for ICU patients via coupled HMM walking on sequential contrast patterns. *Journal of Biomedical Informatics*, *66*, 19–31. Retrieved 2021-07-08, from https://www.sciencedirect.com/science/article/pii/S1532046416301848 doi: 10.1016/j.jbi.2016.12.010

Goh, K. H., Wang, L., Yeow, A. Y. K., Poh, H., Li, K., Yeow, J. J. L. & Tan, G. Y. H. (2021, January). Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. *Nature Communications*, *12*(1), 711. Retrieved 2021-07-25, from https://www.nature.com/articles/s41467-021-20910-4 (Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Infection;Infectious-disease diagnostics;Predictive medicine Subject_term_id: infection;infectious-disease-diagnostics;predictive-medicine) doi: 10.1038/s41467-021-20910-4

Griffin, M. P. & Moorman, J. R. (2001, January). Toward the early diagnosis of neonatal sepsis and sepsis-like illness using novel heart rate analysis. *Pediatrics*, *107*(1), 97–104. doi: 10.1542/peds.107.1.97

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., ... Chen, T. (2017, October). Recent Advances in Convolutional Neural Networks. *arXiv:1512.07108 [cs]*. Retrieved 2021-08-12, from http://arxiv.org/abs/1512.07108 (arXiv: 1512.07108)

Guo, S., Lin, Y., Feng, N., Song, C. & Wan, H. (2019, July). Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*(01), 922–929. Retrieved 2021-07-20, from https://ojs.aaai.org/index.php/AAAI/article/view/3881 (Number: 01) doi: 10.1609/aaai.v33i01.3301922

Gwadry-Sridhar, F., Lewden, B., Mequanint, S. & Bauer, M. (2009, January). COMPARISON OF ANALYTIC APPROACHES FOR DETERMINING VARIABLES - A Case Study in Predicting the Likelihood of Sepsis. In (Vol. 1, pp. 90–96). SCITEPRESS. Retrieved 2021-07-25, from http://www.scitepress.org/documents/2009/15541 doi: 10.5220/0001554100900096

Gül, F., Arslantaş, M. K., Cinel, s. & Kumar, A. (2017, June). Changing Definitions of Sepsis. *Turkish Journal of Anaesthesiology and Reanimation*, *45*(3), 129–138. Retrieved 2021-08-23, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5512390/ doi: 10.5152/TJAR.2017.93753

Henry, K., Hager, D., Pronovost, P. & Saria, S. (2015). A targeted real-time early warning score (TREWScore) for septic shock. *Science Translational Medicine*. doi: 10.1126/scitranslmed.aab3719

Ho, J. C., Lee, C. H. & Ghosh, J. (n.d.). *Imputation-Enhanced Prediction of Septic Shock in ICU Patients.*

Hochreiter, S. & Schmidhuber, J. (1997, November). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. (Conference Name: Neural Computation) doi: 10.1162/neco.1997.9.8.1735

Hu, C.-P. & Chang, Y.-Y. (2017, June). John W. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. *Journal of Social and Administrative Sciences*, *4*(2), 205–207. Retrieved 2021-08-25, from http://www.kspjournals.org/index.php/JSAS/article/view/1313 (Number: 2) doi: 10.1453/jsas.v4i2.1313

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z. & Zhang, H. (2018, October). Deep Learning with Long Short-Term Memory for Time Series Prediction. *arXiv:1810.10161 [cs]*. Retrieved 2021-08-12, from http://arxiv.org/abs/1810.10161 (arXiv: 1810.10161)

Hug, C. (2009). *Detecting Hazardous Intensive Care Patient Episodes Using Real-time Mortality Models* (Thesis). Retrieved 2021-07-25, from https://dspace.mit.edu/handle/1721.1/46690 (Accepted: 2009-08-26T18:15:07Z)

Hunt, E. B., Marin, J. & Stone, P. J. (1966). *Experiments in induction*. Oxford, England: Academic Press. (Pages: xi, 247)

Iwashyna, T. J., Odden, A., Rohde, J., Bonham, C., Kuhn, L., Malani, P., ... Flanders, S. (2014, June). Identifying Patients with Severe Sepsis Using Administrative Claims: Patient-Level Validation of the Angus Implementation of the International Consensus Conference Definition of Severe Sepsis. *Medical care*, *52*(6), e39–e43. Retrieved 2021-07-13, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3568444/ doi: 10.1097/MLR.0b013e318268ac86

Johnson, A. (2021, June). *alistairewj/sepsis3-mimic*. Retrieved 2021-07-16, from https://github.com/alistairewj/sepsis3-mimic (original-date: 2016-06-16T17:50:38Z)

Johnson, A., Pollard, T. & Mark, R. (2015). *MIMIC-III Clinical Database*. PhysioNet. Retrieved 2021-07-08, from https://physionet.org/content/mimiciii/1.4/ (Version Number: 1.4 Type: dataset) doi: 10.13026/C2XW26

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., ... Mark, R. G. (2016, May). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*(1), 160035. Retrieved 2021-07-08, from https://www.nature.com/articles/sdata201635 (Bandiera_abtest: a Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Diagnosis;Health care;Medical research;Outcomes research;Prognosis Subject_term_id: diagnosis;health-care;medical-research;outcomes-research;prognosis) doi: 10.1038/sdata.2016.35

Kam, H. J. & Kim, H. Y. (2017, October). Learning representations for the early detection of sepsis with deep neural networks. *Computers in Biology and Medicine*,

89, 248–255. Retrieved 2021-07-08, from https://www.sciencedirect.com/science/article/pii/S0010482517302743 doi: 10.1016/j.compbiomed.2017.08.015

Kechyn, G., Yu, L., Zang, Y. & Kechyn, S. (2018, March). Sales forecasting using WaveNet within the framework of the Kaggle competition. *arXiv:1803.04037 [cs]*. Retrieved 2021-07-10, from http://arxiv.org/abs/1803.04037 (arXiv: 1803.04037)

Kingma, D. P. & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *ICLR*.

Kotsiantis, S. B. (2013, April). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*(4), 261–283. Retrieved 2021-07-09, from https://doi.org/10.1007/s10462-011-9272-4 doi: 10.1007/s10462-011-9272-4

Kumar, A., Roberts, D., Wood, K. E., Light, B., E., P. J. & et al., S. S. (2006). Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock*. *Critical Care Medicine*, *34*, 1589–1596. doi: https://doi.org/10.1097/01.ccm.0000217961.75225.e9

Lea, C., Vidal, R., Reiter, A. & Hager, G. D. (2016). Temporal Convolutional Networks: A Unified Approach to Action Segmentation. In *Computer Vision – ECCV 2016 Workshops* (Vol. 9915, pp. 47–54). Cham: Springer International Publishing. Retrieved 2021-07-10, from http://link.springer.com/10.1007/978-3-319-49409-8_7 (Series Title: Lecture Notes in Computer Science) doi: 10.1007/978-3-319-49409-8_7

Levy, M. M., Fink, M. P., Marshall, J. C., Abraham, E., Angus, D., Cook, D., ... Ramsay, G. (2003). 2001 sccm/esicm/accp/ats/sis international sepsis definitions conference. *Intensive Care Medicine*, *29*, 530–538. doi: https://doi.org/10.1007/s00134-003-1662-x

Liang, Y., Ke, S., Zhang, J., Yi, X. & Zheng, Y. (2018, July). GeoMAN: Multi-level Attention Networks for Geo-sensory Time Series Prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence* (pp. 3428–3434). Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization. Retrieved 2021-07-20, from https://www.ijcai.org/proceedings/2018/476 doi: 10.24963/ijcai.2018/476

Liao, R., Xiong, Y., Fetaya, E., Zhang, L., Yoon, K., Pitkow, X., ... Zemel, R. (2019, November). Reviving and Improving Recurrent Back-Propagation. *arXiv:1803.06396 [cs, stat]*. Retrieved 2021-08-12, from http://arxiv.org/abs/1803.06396 (arXiv: 1803.06396)

Lin, L., Xu, B., Wu, W., Richardson, T. W. & Bernal, E. A. (n.d.). edical Time Series Classification with Hierarchical Attention-based Temporal Convolutional Networks: A Case Study of Myotonic Dystrophy Diagnosis. , 4.

Liu, J., Yan, J., Wang, L., Huang, L., He, H. & Liu, H. (2021, January). Remote Sensing Time Series Classification Based on Self-Attention Mechanism and Time Sequence Enhancement. *Remote Sensing*, *13*(9), 1804. Retrieved 2021-08-12, from https://www.mdpi.com/2072-4292/13/9/1804 (Number: 9 Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/rs13091804

Liu, Q., Yang, J., Zhang, J., Zhao, F., Feng, X., Wang, X. & Lyu, J. (2019). Description

of Clinical Characteristics of VAP Patients in MIMIC Database. *Frontiers in Pharmacology*, *10*. Retrieved 2021-07-08, from https://www.frontiersin.org/articles/10.3389/fphar.2019.00062/full (Publisher: Frontiers) doi: 10.3389/fphar.2019.00062

Liu, Y., Dong, H., Wang, X. & Han, S. (2019, June). Time Series Prediction Based on Temporal Convolutional Network. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)* (pp. 300–305). doi: 10.1109/ICIS46139.2019.8940265

Lukaszewski, R. A., Yates, A. M., Jackson, M. C., Swingler, K., Scherer, J. M., Simpson, A. J., ... Pearce, M. J. (2008, July). Presymptomatic Prediction of Sepsis in Intensive Care Unit Patients. *Clinical and Vaccine Immunology*, *15*(7), 1089–1094. Retrieved 2021-07-25, from https://journals.asm.org/doi/full/10.1128/CVI.00486-07 (Publisher: American Society for Microbiology) doi: 10.1128/CVI.00486-07

Mao, Q., Jay, M., Hoffman, J. L., Calvert, J., Barton, C. & Shimabukuro, D. e. a. (2018). Multicentre validation of a sepsis prediction algorithm using only vital sign data in the emergency department, general ward and icu. *BMJ Open*, *8*. doi: https://doi.org/10.1136/bmjopen-2017-017833

Margherita. (2021, March). *mmr12/MGP-AttTCN*. Retrieved 2021-07-08, from https://github.com/mmr12/MGP-AttTCN (original-date: 2020-06-07T15:18:46Z)

Mnih, V., Heess, N. & Graves, A. (n.d.). Recurrent Models of Visual Attention. , 9.

Model Evaluation Metrics in Machine Learning. (n.d.). Retrieved 2021-08-12, from https://www.kdnuggets.com/model-evaluation-metrics-in-machine-learning.html/

Moor, M., Horn, M., Rieck, B., Roqueiro, D. & Borgwardt, K. (2019, October). Early Recognition of Sepsis with Gaussian Process Temporal Convolutional Networks and Dynamic Time Warping. In *Machine Learning for Healthcare Conference* (pp. 2–26). PMLR. Retrieved 2021-07-08, from http://proceedings.mlr.press/v106/moor19a.html (ISSN: 2640-3498)

Moorman, J., Lake, D. & Griffin, M. (2006, January). Heart rate characteristics monitoring for neonatal sepsis. *IEEE Transactions on Biomedical Engineering*, *53*(1), 126–132. (Conference Name: IEEE Transactions on Biomedical Engineering) doi: 10.1109/TBME.2005.859810

Mujtaba, H. (2020, September). *An Introduction to Grid Search CV | What is Grid Search*. Retrieved 2021-07-18, from https://www.mygreatlearning.com/blog/gridsearchcv/ (Section: Machine Learning)

Nan, M., Trăscău, M., Florea, A. M. & Iacob, C. C. (2021, January). Comparison between Recurrent Networks and Temporal Convolutional Networks Approaches for Skeleton-Based Action Recognition. *Sensors*, *21*(6), 2051. Retrieved 2021-08-12, from https://www.mdpi.com/1424-8220/21/6/2051 (Number: 6 Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/s21062051

Nguyen, H. B., Corbett, S. W., Steele, R. & Banta, J. e. a. (2007). Implementation

of a bundle of quality indicators for the early management of severe sepsis and septic shock is associated with decreased mortality*. *Critical Care Medicine*, *35*, 1546–1554. doi: https://doi.org/10.1097/01.ccm.0000259463.33848.3d

Oh, S. L., Jahmunah, V., Ooi, C. P., Tan, R. S., Ciaccio, E. J., Yamakawa, T., ... Rajendra Acharya, U. (2020). Classification of heart sound signals using a novel deep wavenet model. *Computer Methods and Programs in Biomedicine*, *196*. doi: https://doi.org/10.1016/j.cmpb.2020.105604

Olanow, C. W., Watts, R. L. & Koller, W. C. (2001, June). An algorithm (decision tree) for the management of Parkinson's disease (2001): treatment guidelines. *Neurology*, *56*(11 Suppl 5), S1–S88. doi: 10.1212/wnl.56.suppl_5.s1

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016, September). WaveNet: A Generative Model for Raw Audio. *arXiv:1609.03499 [cs]*. Retrieved 2021-07-10, from http://arxiv.org/abs/1609.03499 (arXiv: 1609.03499)

Otero, R. M., Nguyen, H. B., Huang, D. T., Gaieski, D. F., Goyal, M., Gunnerson, K. J., ... Rivers, E. P. (2006, November). Early Goal-Directed Therapy in Severe Sepsis and Septic Shock Revisited: Concepts, Controversies, and Contemporary Findings. *Chest*, *130*(5), 1579–1595. Retrieved 2021-07-25, from https://www.sciencedirect.com/science/article/pii/S0012369215373402 doi: 10.1378/chest.130.5.1579

Paetz, J. (2003, June). Knowledge-based approach to septic shock patient data using a neural network with trapezoidal activation functions. *Artificial Intelligence in Medicine*, *28*(2), 207–230. Retrieved 2021-07-25, from https://www.sciencedirect.com/science/article/pii/S0933365703000575 doi: 10.1016/S0933-3657(03)00057-5

pgAdmin - PostgreSQL Tools. (n.d.). Retrieved 2021-07-18, from https://www.pgadmin.org/

Pires, H. H. G., Neves, F. F. & Pazin-Filho, A. (2019, November). Triage and flow management in sepsis. *International Journal of Emergency Medicine*, *12*(1), 36. Retrieved 2021-08-23, from https://doi.org/10.1186/s12245-019-0252-9 doi: 10.1186/s12245-019-0252-9

Python utilized LSTM time series prediction model analysis(Others-Community). (n.d.). Retrieved 2021-08-12, from https://titanwolf.org/Network/Articles/Article?AID=09b887d6-9b8d-46f0-ba7a-a3b497ecc124#gsc.tab=0

Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G. & Cottrell, G. (2017, August). A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *arXiv:1704.02971 [cs, stat]*. Retrieved 2021-07-08, from http://arxiv.org/abs/1704.02971 (arXiv: 1704.02971)

Raghu, A., Komorowski, M. & Singh, S. (2018, November). Model-Based Reinforcement Learning for Sepsis Treatment. *arXiv:1811.09602 [cs, stat]*. Retrieved 2021-07-07, from http://arxiv.org/abs/1811.09602 (arXiv: 1811.09602)

Rather, A. R. & Kasana, B. (2015). The third international consensus definitions for

sepsis and septic shock (sepsis-3). *JMS SKIMS*, *18*, 162–164. doi: https://doi.org/10.33883/jms.v18i2.269

Safavian, S. & Landgrebe, D. (1991, May). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(3), 660–674. (Conference Name: IEEE Transactions on Systems, Man, and Cybernetics) doi: 10.1109/21.97458

Schlemper, J., Oktay, O., Schaap, M., Heinrich, M., Kainz, B., Glocker, B. & Rueckert, D. (2019, April). Attention gated networks: Learning to leverage salient regions in medical images. *Medical Image Analysis*, *53*, 197–207. Retrieved 2021-07-08, from https://www.sciencedirect.com/science/article/pii/S1361841518306133 doi: 10.1016/j.media.2019.01.012

Scikit-Learn - Model Evaluation & Scoring Metrics. (n.d.). Retrieved 2021-08-12, from https://coderzcolumn.com/tutorials/machine-learning/model-evaluation-scoring-metrics-scikit-learn-sklearn

Shavdia, D. (2007). *Septic shock : providing early warnings through multivariate logistic regression models* (Thesis, Massachusetts Institute of Technology). Retrieved 2021-07-08, from https://dspace.mit.edu/handle/1721.1/42338 (Accepted: 2008-09-03T15:23:00Z)

Sheetrit, E., Nissim, N., Klimov, D., Fuchs, L., Elovici, Y. & Shahar, Y. (2017, September). Temporal Pattern Discovery for Accurate Sepsis Diagnosis in ICU Patients. *arXiv:1709.01720 [cs, stat]*. Retrieved 2021-07-17, from http://arxiv.org/abs/1709.01720 (arXiv: 1709.01720)

Sherstinsky, A. (2020, March). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Physica D: Nonlinear Phenomena*, *404*, 132306. Retrieved 2021-08-12, from http://arxiv.org/abs/1808.03314 (arXiv: 1808.03314) doi: 10.1016/j.physd.2019.132306

Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., ... Angus, D. C. (2016, February). The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). *JAMA*, *315*(8), 801–810. Retrieved 2021-07-08, from https://doi.org/10.1001/jama.2016.0287 doi: 10.1001/jama.2016.0287

Tang, C. H. H., Middleton, P. M., Savkin, A. V., Chan, G. S. H., Bishop, S. & Lovell, N. H. (2010, May). Non-invasive classification of severe sepsis and systemic inflammatory response syndrome using a nonlinear support vector machine: a preliminary study. *Physiological Measurement*, *31*(6), 775–793. Retrieved 2021-07-25, from https://doi.org/10.1088/0967-3334/31/6/004 (Publisher: IOP Publishing) doi: 10.1088/0967-3334/31/6/004

Tawfik, G. M., Dila, K. A. S., Mohamed, M. Y. F., Tam, D. N. H., Kien, N. D., Ahmed, A. M. & Huy, N. T. (2019, August). A step by step guide for conducting a systematic review and meta-analysis with simulation data. *Tropical Medicine and Health*, *47*(1), 46. Retrieved 2021-08-25, from https://doi.org/10.1186/s41182-019-0165-6 doi: 10.1186/s41182-019-0165-6

Thiel, S. W., Rosini, J. M., Shannon, W., Doherty, J. A., Micek, S. T. & Kollef, M. H. (2010). Early prediction of septic shock in hospitalized patients. *Journal of Hospital Medicine*, *5*(1), 19–25. Retrieved 2021-07-08, from https://onlinelibrary.wiley.com/doi/abs/10.1002/jhm.530 (_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jhm.530) doi: 10.1002/jhm.530

Toma, T., Abu-Hanna, A. & Bosman, R.-J. (2007, December). Discovery and inclusion of SOFA score episodes in mortality prediction. *Journal of Biomedical Informatics*, *40*(6), 649–660. doi: 10.1016/j.jbi.2007.03.007

Vieira, S. M., Mendonça, L. F., Farinha, G. J. & Sousa, J. M. C. (2013, August). Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Applied Soft Computing*, *13*(8), 3494–3504. Retrieved 2021-07-25, from https://www.sciencedirect.com/science/article/pii/S1568494613001361 doi: 10.1016/j.asoc.2013.03.021

Vincent, J. L. (1997, February). Dear SIRS, I'm sorry to say that I don't like you.. *Critical Care Medicine*, *25*(2), 372–374. doi: 10.1097/00003246-199702000-00029

Vincent, J. L., de Mendonça, A., Cantraine, F., Moreno, R., Takala, J., Suter, P. M., ... Blecher, S. (1998, November). Use of the SOFA score to assess the incidence of organ dysfunction/failure in intensive care units: results of a multicenter, prospective study. Working group on "sepsis-related problems" of the European Society of Intensive Care Medicine. *Critical Care Medicine*, *26*(11), 1793–1800. doi: 10.1097/00003246-199811000-00016

Vincent, J. L., Marshall, J. C., Ñamendys Silva, S. A. & François, B. (2014). Assessment of the worldwide burden of critical illness: the intensive care over nations (icon) audit. *The Lancet Respiratory Medicine*, *2*(5), 380–386.

Vincent, J. L., Moreno, R., Takala, J., Willatts, S., De Mendonça, A., Bruining, H., ... Thijs, L. G. (1996). The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure. *Intensive Care Medicine*, *22*, 707–710. doi: https://doi.org/10.1007/bf01709751

Williams, C., Bonilla, E. & Chai, M. A. (2008). *Multi-task gaussian process prediction.* Institute for Adaptive and Neural Computation school of Informatics University of Edinburgh, UK.

Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. (2018, July). CBAM: Convolutional Block Attention Module. *arXiv:1807.06521 [cs]*. Retrieved 2021-07-08, from http://arxiv.org/abs/1807.06521 (arXiv: 1807.06521)

Xie, Y., Liang, R., Liang, Z., Huang, C., Zou, C. & Schuller, B. (2019, November). Speech Emotion Classification Using Attention-Based LSTM. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(11), 1675–1685. (Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing) doi: 10.1109/TASLP.2019.2925934

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. & Hovy, E. (2016, June). Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies (pp. 1480–1489). San Diego, California: Association for Computational Linguistics. Retrieved 2021-07-08, from https://aclanthology.org/N16-1174 doi: 10.18653/v1/N16-1174

Yildirim, O., Baloglu, U. B., Tan, R. S., Ciaccio, E. J. & Acharya, U. R. (2001). A new approach for arrhythmia classification using deep coded features and lstm networks. *Computer Methods and Programs in Biomedicine*, *176*, 121–133. doi: 10.1016/j.cmpb.2019.05.004

Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M. & Berg, T. L. (2018). MAttNet: Modular Attention Network for Referring Expression Comprehension. In (pp. 1307–1315). Retrieved 2021-07-08, from https://openaccess.thecvf.com/content_cvpr_2018/html/Yu_MAttNet_Modular_Attention_CVPR_2018_paper.html

Zhou, Z.-H. & Chen, Z.-Q. (2002, November). Hybrid decision tree. *Knowledge-Based Systems*, *15*(8), 515–528. Retrieved 2021-07-09, from https://www.sciencedirect.com/science/article/pii/S0950705102000382 doi: 10.1016/S0950-7051(02)00038-2

Appendix A

Glossary

TCN Temporal Convolutional Networks. TCN is a new type of algorithm that can be used to solve time series forecasting. It can take a series of arbitrary lengths and output them as the same length. In the case of using a one-dimensional fully convolutional network architecture, causal convolution is used.

MIMIC-III Medical Information Mart for Intensive Care. It is a critical care medicine database. MIMIC III has collected data on cases hospitalized in Beth Israel Deaconess Medical Center for more than 10 years. These data have been sorted into csv data, a total of 26 tables, which we can import into a database or other tools to view.

ICD-9 code ICD (International Classification of Diseases) is a system that classifies diseases according to certain characteristics and rules, and uses coding methods to represent them. The ninth revision is ICD-9.

LSTM Long Short-Term Memory. It is a kind of RNN that is capable of learning long-term dependent information..

Benchmark Model In order to discuss the influence of the factors of concern on the equilibrium, we usually need a benchmark model. In this paper, we are concerned about the advantages of the TCN model, so we established two benchmark models:

decision tree and LSTM model.

AUC-ROC Area Under Curve. AUC-ROC is to reflect the TP rate (TPR) and FP rate (FPR) obtained under different thresholds.

SOFA Sequential Organ Failure Assessment. Dynamic detection of $SOFA$ score can better reflect the degree of organ damage and treatment effect in critically ill patients, and has a very reliable guiding significance for key treatment. When the $SOFA\ score$ >= 2, it can be considered that the patient has OD (organ failure), that is to say $Spesis3.0\ =\ Infection$ + $SOFA$ >= 2, this paper adopts Spesis3.0 version.

Machine learning Machine learning is a phrase that refers to a group of algorithms that seek to extract hidden rules from huge amounts of historical data and apply them to prediction or categorization. Machine learning models are divided into regression and classification models. This paper establishes a decision tree classification model.

Deep learning First here, input is the known information, and the output is the final result of cognition. The explanation of "learning" is the process of obtaining cognition from existing information through calculation, judgment and reasoning. The learning effect is different because of the different learning strategies. The "neural network" is constructed by academia imitating the neural network of the human brain. The interpretation of "depth" is the depth of the hidden layer, and the total number of layers experienced from the "input layer" to the "output layer" is the depth of the neural network. The more hidden layers, the deeper the neural network.

MGP | MTGP multi-task Gaussian processes. The Gaussian process uses the probability distribution to represent the prior knowledge of the function output, and builds a model in the functional space. It constructs a covariance function based on the correlation between data, and calculates it through Bayesian inference.

MTGP is an extension of GP. Its key is to identify the correlation of multiple outputs by using the following new covariance kernel function:

$$K_{MTGP}(i, i', s, s') = k_c(s, s') * k_t(x, x')$$

$s, s' \in \{1, 2, ..., m\}$ represents the index of the two sequences.

k_c and k_t respectively model the relationship between multiple outputs and the covariance in a single sequence.

i and i' represent the time index of tasks s and s'.

Appendix B

Additional information here

B.1 Appendix Arrangements

The contents of each appendix are arranged as follows

Appendix A: Glossary

Appendix B: Appendix Arrangements

Appendix C: Ethics

a) Data Ethics

b) Ethics Approval

Appendix B: Tools

a) PostgreSQl,pgAdmin4

b) python

c) scala,Intellij IDEA

Appendix C

Ethics

C.1 Data Ethics

In the MIMIC-III database, subject_id is used to represent the identity of the patient. Sensitive information such as the patient's name and unique identification number is hidden. That is, the MIMIC-III database itself has processed information related to personal privacy, which meets the ethical requirements of privacy protection.

Looking at the time recorded in the table in the MIMIC-III database, we find that there is a year like "2181". This is because MIMIC-III does not directly use real time due to data ethics requirements. It offsets the time and randomly adds or subtracts some ancient numbers. Events handled include time of admission, time of birth, time of death, etc.

C.2 Ethics Approval

In order to gain access to the MIMIC-III database, I first registered on the physionet website (registration URL: `https://physionet.org/pnw/login`) and filled in my real information. Then I took the ethics exam on the CITI website (URL:

Figure C.1: CITI Ethics Exam Pass Certificate

`https://about.citiprogram.org/en/homepage/`) and passed the exam. CITI issued me a certificate of qualification (see Figure C.1). Then upload the certificate on physionet to get the MIMIC permission. So I use the MIMIC-III database because the data is ethically approved, and I use it for academic research without commercial behavior. After obtaining the database usage permission, the next thing I have to do is to download the data to the local computer, and use the Postgres software to install and import.

Appendix D

Tools

D.1 Database Tools

PostgreSQL Install postgreSQL database, import MIMIC-III database

pgAdmin4 pgAdmin 4 is a reliable and comprehensive database design and management software designed for PostgreSQL. It allows you to connect to a specific database, create tables and run various SQL statements from simple to complex.

D.2 Code Tools

python a) Environment: Google Golab

b) Libraries: scikit-learn and pytorch

Scala Intellij IDEA